

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Toshihiro SUZUKI

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HEREWITH

FOR: CHARACTER RECOGNITION METHOD AND COMPUTER-READABLE STORAGE MEDIUM

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS

WASHINGTON, D.C. 20231

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §119(e).
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

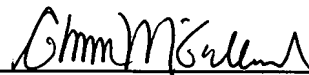
<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
JAPAN	2000-094959	March 30, 2000

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
(B) Application Serial No.(s)
 - ☐ are submitted herewith
 - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

C. Irvin McClelland
Registration Number 21,124



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/98)



日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

#2
J1046 U.S. PRO
09/819866



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 3月30日

出願番号

Application Number:

特願2000-094959

出願人

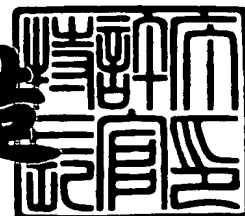
Applicant (s):

株式会社リコー

2001年 3月 2日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3013420

【書類名】 特許願
 【整理番号】 9908806
 【提出日】 平成12年 3月30日
 【あて先】 特許庁長官 殿
 【国際特許分類】 G06K 9/34
 【発明の名称】 文字認識方法および記録媒体
 【請求項の数】 17
 【発明者】
 【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社 リコー
 内
 【氏名】 鈴木 俊博
 【特許出願人】
 【識別番号】 000006747
 【氏名又は名称】 株式会社 リコー
 【代表者】 桜井 正光
 【代理人】
 【識別番号】 100073760
 【弁理士】
 【氏名又は名称】 鈴木 誠
 【選任した代理人】
 【識別番号】 100097652
 【弁理士】
 【氏名又は名称】 大浦 一仁
 【手数料の表示】
 【予納台帳番号】 011800
 【納付金額】 21,000円
 【提出物件の目録】
 【物件名】 明細書 1
 【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9809191

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 文字認識方法および記録媒体

【特許請求の範囲】

【請求項 1】 文字画像の特徴として断面系列グラフを用いて文字認識する文字認識方法であって、文字列画像から断面系列グラフを抽出し、該断面系列グラフの特異領域を解析し、該解析結果を基に前記特異領域に仮想境界点列を生成し、前記断面系列グラフの構造要素を組み合わせて文字候補を生成し、該生成された文字候補について、必要な仮想境界点列を補ってから一文字として認識し、一文字認識された文字候補間の隣接関係を基に文字列を認識することを特徴とする文字認識方法。

【請求項 2】 前記仮想境界点列が生成される特異領域は、文字列が近接あるいは接触している領域を含むことを特徴とする請求項 1 記載の文字認識方法。

【請求項 3】 前記特異領域と該特異領域に接続する断面系列の接続順序と位置とを基に、前記仮想境界点列の生成位置を決定することを特徴とする請求項 1 または 2 記載の文字認識方法。

【請求項 4】 前記仮想境界点列を生成するとき文字列の方向を参照し、該文字列の方向に対応しない断面系列間には仮想境界点列を生成しないことを特徴とする請求項 1、2 または 3 記載の文字認識方法。

【請求項 5】 前記仮想境界点列は、所定の曲線発生方法を用いて生成することを特徴とする請求項 1、2 または 3 記載の文字認識方法。

【請求項 6】 前記構造要素を第 1 の層と第 2 の層に分け、それぞれをタグで管理することを特徴とする請求項 1 記載の文字認識方法。

【請求項 7】 前記構造要素の第 1 の層は、断面系列と特異領域であり、前記断面系列を第 1 のタグで管理し、前記特異領域を第 2 のタグで管理することを特徴とする請求項 1 または 6 記載の文字認識方法。

【請求項 8】 前記構造要素の第 2 の層は境界点列であり、該境界点列を第 3 のタグで管理することを特徴とする請求項 1 または 6 記載の文字認識方法。

【請求項 9】 前記文字列間の空白領域を第 4 のタグで管理することを特徴とする請求項 1 記載の文字認識方法。

【請求項 1 0】 前記第 1、第 2、第 3 および第 4 のタグを組み合わせることにより、文字候補を生成し、該生成された文字候補を第 5 のタグで管理することを特徴とする請求項 7、8 または 9 記載の文字認識方法。

【請求項 1 1】 前記第 1、第 2 および第 3 のタグを組み合わせるとき、同一の構造要素については、第 1 の層のタグまたは第 2 の層のタグの何れかを用い、タグの重複を解消することを特徴とする請求項 1 0 記載の文字認識方法。

【請求項 1 2】 前記第 5 のタグ間にリンクを生成し、該生成されたリンク間にパスを生成し、該生成されたパスの内、最適なパスを選択することにより、文字列を認識出力することを特徴とする請求項 1 0 記載の文字認識方法。

【請求項 1 3】 文字列画像から断面系列グラフの第 1 の層である断面系列と特異領域を抽出し、断面系列グラフの第 2 の層である境界点列を抽出する機能と、前記断面系列と特異領域および境界点列を組み合わせることにより文字候補を生成する機能と、該生成された文字候補を一文字として認識する機能と、一文字認識された文字候補間の隣接関係を基に文字列を認識する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 1 4】 前記特異領域を解析し、該解析結果を基に前記特異領域に仮想境界点列を生成する機能と、前記一文字を認識する際に、断面系列と特異領域からなる文字候補を、必要な仮想境界点列で補って境界点列に変換する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な請求項 1 3 記載の記録媒体。

【請求項 1 5】 さらに、前記断面系列を管理する第 1 のタグ、前記特異領域を管理する第 2 のタグ、前記境界点列を管理する第 3 のタグ、前記文字列間の空白を管理する第 4 のタグを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な請求項 1 3 または 1 4 記載の記録媒体。

【請求項 1 6】 前記第 1、第 2、第 3 および第 4 のタグを組み合わせることにより生成される文字候補を管理する第 5 のタグを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な請求項

1 5 記載の記録媒体。

【請求項 1 7】 前記第 1、第 2、第 3 および第 5 のタグを、同一の論理構造で管理する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な請求項 1 5 または 1 6 記載の記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、文字認識において、一文字を認識する機能と、接触文字や分離文字等を適切に切り出して認識する文字列認識機能とを有する文字認識方法および文字認識処理プログラムを記録した記録媒体に関し、例えば複数行の文字列認識や罫線接触などを考慮した文字列認識などに適用される。

【0 0 0 2】

【従来技術】

文字列を認識する従来技術として、例えば特開平 5 - 2 3 3 8 7 7 号公報に記載された単語読み取り方法がある。この方法では、文字が極端に分離していたりあるいは接触している場合でも、複数の切り出し方法によって文字を切り出すことにより、正しい文字パターンが得られるようにしている。

【0 0 0 3】

上記した方法では、文字を切り出す際に、文字列パターンの周辺分布のデータと、外接矩形のデータを用いている。つまり、周辺分布を基に矩形を垂直に分割している。また、従来文字認識方法では、文字候補を処理するために画像データに直接アクセスする構成が採られていた。

【0 0 0 4】

【発明が解決しようとする課題】

近接文字列あるいは接触文字列には、互いに矩形状に分離できないもの、あるいは直線的に分割できないものがある。上記した従来方法は、このような近接文字列あるいは接触文字列についても矩形を単位として認識処理しているので、認識精度上の限界がある。

【0 0 0 5】

また、画像データにアクセスする必要があるため、処理時間の増大が避けられない。

【 0 0 0 6 】

本発明は、上記した問題点に鑑みてなされたもので、

本発明の目的は、仮想境界点列を用いることにより、近接あるいは接触文字列を適切に切り出すと共に、処理時間の増大を抑制しつつ、認識精度を向上させた文字認識方法および記録媒体を提供することにある。

【 0 0 0 7 】

本発明の他の目的は、異なる階層の画像特徴をタグで管理することにより、認識処理の高速化を図った文字認識方法および記録媒体を提供することにある。

【 0 0 0 8 】

【課題を解決するための手段】

本発明では、断面系列グラフの特異領域に対して仮想境界点列を生成することにより、接触文字の分離を可能とする。分離文字は仮想境界点列と境界点列で表されて文字認識される。上記した仮想境界点列は、滑らかな曲線で作成され、文字認識の認識精度を向上させる。

【 0 0 0 9 】

また、本発明では、断面系列グラフの構造要素をタグで管理すると共に、タグの表現形式（論理構造）を共通化する。

【 0 0 1 0 】

【発明の実施の形態】

以下、本発明の一実施例を図面を用いて具体的に説明する。

まず、本発明の原理について説明する。本発明の基本となる考え方は、以下の3つである。

【 0 0 1 1 】

その第1は、「特徴要素（後述するSSタグ、SRタグ、BCCタグ）を組み合わせる」という考え方である。これは断面系列グラフの利用および仮想境界点列という新しい手法によって実現される。この手法により、接触文字列を正しく切り出すことで認識精度を向上させ、またその際の処理量の増大を抑制する。

【 0 0 1 2 】

第2は、「複数種類の文字要素（断面系列、特異領域、境界点列）と文字候補とに関する表現形式の共通化」という考え方であり、異なる階層の特徴（断面系列グラフの上位層と下位層）をタグで管理し、タグの表現形式を共通化するという新しい手法により実現される。これにより、処理量が圧縮され、特に処理手段が簡易化される。

【 0 0 1 3 】

第3は、「階層性を利用した、バックトラック型アプローチと仮説検証型（パス選択型）アプローチの融合」という考え方である。両者はよく知られている既存の方法である。簡単に言えば、前者は、「怪しい場合に限って処理をやり直す」というアプローチである。後者は、「始めからより多くの可能性を列挙しておいて、それらの組み合わせのなかで最適なものを選択する」というアプローチである。これらの融合は、上記したタグの導入に加え、そこでの予約・継承処理という新しい手法により実現される。これにより、認識精度を維持しつつ処理量の増大が抑制される。

【 0 0 1 4 】

すなわち、本発明は、「断面系列グラフに仮想境界点列を加えて、タグで管理すること」によって、認識精度と処理量とバランスを図っている。

【 0 0 1 5 】

（断面系列グラフ）

まず、断面系列グラフについて説明する。断面系列グラフとは、本発明者によって提案された線図形を表現する方法である（特許第2615247号および特開平6-68305号公報を参照）。

【 0 0 1 6 】

すなわち、線図形イメージデータから線分の方角にほぼ直交する断面を抽出し、この断面の連なりを断面系列とし、また断面系列以外の領域を不明領域とし、ノードで断面系列または不明領域を表わすとともにアークでノードの接続関係を表わしたグラフ構造（断面系列グラフ）を作成し、この断面系列グラフを線図形の特徴とすることを骨子とするものである。

【 0 0 1 7 】

図 1 2 は、線図形イメージデータ（原画像データ）と断面系列グラフ並びに骨格線（芯線）との関係を示す。このような階層構造を持つ断面系列グラフによる線分記述において、その構造要素である断面は線分の方向にほぼ直交するという性質を持ち、またその上位の構造要素である断面系列は、断面をあたかも時系列のように順序付けたものとなっている（図 1 2 の（イ）の部分）。不明領域は、断面系列に含まれない領域であって、線端や屈曲、分岐あるいは交差の面状の領域がこれに相当する（図 1 2 の（ロ）の部分）。

【 0 0 1 8 】

このように断面系列グラフは、線図形を単純な線とそうでない部分に明確に分けて線図形を記述したシンプルなものであり、これに基づいて骨格線等の構造的特徴マッチングや大まかな構造的マッチングをとることが容易になる。

【 0 0 1 9 】

断面系列グラフは、画像データを階層的に記述するものである。図 1 3 は、文字列画像の例であり、図 1 4 は、これに対する断面系列グラフの境界点と断面を示し、図 1 5 は、断面系列グラフの断面系列と特異領域を示す。

【 0 0 2 0 】

図 1 4 において、輪郭上の各点（黒画素側）を境界点と呼ぶ。また、輪郭に相当する境界点の並びを境界点列と呼ぶ。境界点列の種類には、外側と内側とがあり、図 1 4 の例では、1 つの外側境界点列と、2 つの内側境界点列（図 1 4 の“8”という文字の内側にある“○”と“○”部分）がある。

【 0 0 2 1 】

また、断面とは、線分の方向にほぼ直交するように求めた境界点ペアのことであり、図 1 4 では境界点を結ぶ直線で表現されている。断面を隙間なく矛盾なく並べた一連の並びが断面系列であり、それ以外が特異領域である。特異領域は、前掲した特許第 2 6 1 5 2 4 7 号では、不明領域に相当している。本発明では特異領域という。断面系列が通常の文字線部を表し、特異領域がその接続部や端点部を表している。

【 0 0 2 2 】

{境界点、境界点列、断面、断面系列、特異領域} という構造要素は、それら自体が特徴であるとともに、それらを組み合わせることによって、画像データを構造化する役割も持っている。

【 0 0 2 3 】

{境界点、境界点列、断面} を下位層の構造要素とし、{断面系列、特異領域} を上位層の構造要素とすることにより、上位層は線図形（文字）の大まかな構造を表し、下位層はより詳細な構造を表している。

【 0 0 2 4 】

ただし、本発明においては、後述する抽出ステップのエラー対策などを考慮して、断面という構造要素を、下位層ではなく上位層に位置づけて説明する。

【 0 0 2 5 】

本発明では、断面系列グラフを利用することを基本としているが、その具体的な表現方法を図 1 6 ～ 2 0 に示す。図 1 6 ～ 2 0 は、それぞれの構造要素の記述形式と抽出方法とを示したものである。抽出方法については、前掲した特許第 2 6 1 5 2 4 7 号等に記載にされているので省略する。

【 0 0 2 6 】

図 1 6 は、境界点の表現形式を示す。1 つの境界点は、1 つの構造体で記述される。その構造体は、具体的には、図 1 6 の右に示すように、メモリ上の連続領域 1 8 に配置され、この連続領域 1 8 は、その上限を示す M_b と、実際に使われている境界点数 N_b の 2 つの変数によって管理される。

【 0 0 2 7 】

境界点構造体 1 0 は、座標値 (x 、 y) 1 1、濃淡傾き方向 (D_x 、 D_y) 1 2 (文字背景から文字線部に向かう方向であり、境界方向とほぼ垂直)、断面系列番号 1 3、境界点列番号 1 4、前境界点の番号 1 5 (文字線部を左に見て次に現れる境界点の番号)、後境界点の番号 1 6 (文字線部を左に見て、前に現れた境界点の番号)、対向境界点の番号 1 7 (注目境界点 p から濃淡傾き方向に黒画素を探索して最初に現れた境界点 q を、 p に対する対向境界点と呼び、前掲した特許で延長境界点と呼ぶもの) からなる。

【 0 0 2 8 】

図 1 7 は、境界点列の表現形式を示す。前述した境界点構造体と同様に、1 つの境界点列は、1 つの構造体で記述され、複数の境界点列 $0 \sim M_b - 1$ がメモリ上の連続領域 2 7 に配置されている。

【 0 0 2 9 】

境界点列構造体 2 0 は、ステータス 2 1（外側境界点列、内側境界点列、ノイズを区別するフラグ）、最上左の境界点番号 2 2、境界点数 2 3（一周するときの境界点数）、境界点列の外接矩形の左上座標値（ X_s 、 Y_s ）2 4、境界点列の外接矩形の右下外接座標値（ X_e 、 Y_e ）2 5、ワークフラグ 2 6（様々な処理で一時的に使用されるワーク変数）からなる。

【 0 0 3 0 】

図 1 8 は、断面の表現形式を示す。前述した構造体と同様に、1 つの断面は、1 つの構造体で記述され、複数の断面 $0 \sim M_s - 1$ がメモリ上の連続領域 3 5 に配置されている。

【 0 0 3 1 】

断面構造体 3 0 は、一方の境界点の番号（ b_1 ）3 1、他方の境界点の番号（ b_2 ）3 2、 b_1 の前方向側の断面の番号 3 3、 b_2 の後方向側の断面の番号 3 4 からなる。

【 0 0 3 2 】

図 1 9 は、断面系列の表現形式を示す。断面系列構造体 4 0 は、ステータス 4 1（有効／無効を示すフラグ）、断面数 4 2、Head 側の断面番号 4 3（断面系列のある断面が、前方向に別の断面を持たないとき、その断面を断面系列の Head と呼ぶ）、Tail 側の断面番号 4 4（ある断面が後方向に断面を持たないとき、その断面を断面系列の Tail と呼ぶ）、Head 側の特異領域番号 4 5、Tail 側の特異領域番号 4 6、平均断面長 4 7（推定線幅に相当）、その他のワーク変数 4 8（グラフ表現の点（ノード）番号など）からなる。

【 0 0 3 3 】

図 2 0 は、特異領域の表現形式を示し、特異領域構造体 5 0 は、ステータス 5 1（有効／無効を示すフラグ）、接続する断面系列の数 5 2（以下の 2 つの配列の有効数）、断面系列番号の列 5 3、接続タイプの列 5 4（列 5 3 と列 5 4 は互

いに対応する配列であり、タイプとは断面系列の端を区別するものである）、外接座標値（ X_s 、 Y_s ）55、外接座標値（ X_e 、 Y_e ）56、その他のワーク変数57（グラフ表現の点（ノード）番号など）からなる。

【0034】

上記した構造体の断面系列の番号、境界点列の番号などを介して、他の構造体にアクセスすることが可能となる。

【0035】

断面系列グラフは、前掲した特許に記載されているように、それを求める過程で、骨格のグラフ表現が抽出される。また、境界点列を多角形近似することにより、輪郭のグラフ表現を抽出することも可能である。これらのグラフ表現を解析する方法と併用することで、一文字を認識する手段を実現することができる。また、境界点の方向性などを統計的に解析する方法と併用することにより、異なる一文字認識手段を実現することができる。

【0036】

以下の説明では、上記した構造要素を次のように表記する。境界点をB（Borderの略）、境界点列をC（Contourの略）、断面をS（Sliceの略）、断面系列をSS（Slice Sequenceの略）、特異領域をSR（Singular Regionの略）で表す。

【0037】

（仮想境界点列について）

本発明の一つの特徴は、仮想境界点列という考え方をを用いる点にある。これを図を用いて説明する。

【0038】

図21は、境界点 b_s から境界点 b_e へとつながる仮想境界点列セグメントを示す。仮想境界点列セグメントの各境界点は黒丸で示され、これらは図16と同一の表現形式で記述される。図16の境界点構造体10には、前境界点の番号15と後境界点の番号16とがあるが、これらを用いて境界点構造体は双方向リスト状に接続される。

【0039】

本来は、開始境界点 $b s$ と $b s$ の次境界点 $b s R$ (R は $R e a l$ の略) が接続されていたものを、この接続を切り替えて、 $b s$ と $b s$ の仮想次境界点 $b s V$ (V は $V i r t u a l$ の略) が接続されているようにすれば、あたかも電車のポイント切換えを行ったかのようになる。終了境界点 $b e$ 側でも同様につなぎ換えた上で、境界点を辿ったならば、得られる境界点列 (広義の仮想境界点列) は、元のものとは異なったものとなる。この仮想境界点列は、後述するように、接触文字の分離を表現する際に、有効に機能する。

【 0 0 4 0 】

仮想境界点列セグメントとそのつなぎ換え情報とを合わせて、狭義の仮想境界点列と呼び、仮想境界点列は図 2 2 の表現形式で記述される。構造体の管理方法は、図 1 6 ～ 2 0 で説明した方法と同様である。

【 0 0 4 1 】

図 2 2 に示す仮想境界点列構造体 6 0 は、後述する (図 2 3 に示す) キー 6 1、境界点数 6 2、図 2 1 で説明した開始境界点番号 ($b s$) 6 3、 $b s$ の次境界点番号 ($b s R$) 6 4、 $b s$ の仮想次境界点番号 ($b s V$) 6 5 (これら 3 つが開始境界点の接続換え情報に相当)、終了境界点番号 ($b e$) 6 6、 $b e$ の前境界点番号 ($b e R$) 6 7、 $b e$ の仮想前境界点番号 ($b e V$) 6 8 (これら 3 つが終了境界点の接続換え情報に相当) から構成されている。

【 0 0 4 2 】

キー 6 1 は、求められた仮想境界点列を、後で検索するための情報である。キーの形式は、図 2 3 に示すように、特異領域番号 6 1 1 と、開始境界点 ($b s$) を持つ断面系列の参照番号 6 1 2 と、終了境界点 ($b e$) を持つ断面系列の参照番号 6 1 3 からなり、特異領域の接続パターンを表したものである。なお、仮想境界点列を動的に求める場合には、このキーを省略することができる。

【 0 0 4 3 】

図 2 3 に示す接続パターン例は、図 2 4 の特異領域 ($S R 5$) を対象としている。いま、特異領域 $S R 5$ と断面系列 $S S 9$ (下側)、 $S S 6$ (左側) が指定された場合の接続パターンを考える。 $S R 5$ から 2 つの断面系列を参照する場合、まず図 2 0 に示す特異領域番号 ($S R 5$) の特異領域構造体 5 0 が参照され、次

いで配列データ 5 3 が参照される。この配列データは、反時計回りの順序になるようにあらかじめ設定されている。

【 0 0 4 4 】

すなわち、具体的には、配列の 2 番目に S S 6 がセットされていて、3 番目に S S 9 がセットされているとする。この場合、接続パターンの表現は図 2 3 に示すようになる。

【 0 0 4 5 】

ところで、断面系列がループを構成して同一の特異領域に接続する場合に問題が生じる。すなわち、図 2 4 の例では、S S 6 の代わりに S S 9 の他方がある場合などでは、単純に断面系列番号のみで指定すると、4 つの境界点のうち、どれを開始点とし、どれを終了点とすればよいかを特定できなくなる。

【 0 0 4 6 】

そこで、どのようにして仮想境界点列セグメントの開始点と終了点を特定するかについて、以下説明する。

【 0 0 4 7 】

図 2 4 の例では、特異領域 S R 5 に対して、4 つの断面系列 { S S 3, S S 6, S S 7, S S 9 } が接続している。

【 0 0 4 8 】

S R 5 から断面系列を参照するための情報は、図 2 0 の配列データ 5 3 に相当する。これを、S R 5. S S [i] (i = 0, . . . , 3) と表記する。これは、前述したように、S R 5 を左側に見るようにして反時計周りに断面系列を並べるときの順序に一致しているものとする。

【 0 0 4 9 】

つまり、図 2 4 の場合は、

S R 5. S S [0] = S S 7, S R 5. S S [1] = S S 3, S R 5.
S S [2] = S S 6, S R 5. S S [3] = S S 9
と表現される。

【 0 0 5 0 】

さて、指定された断面系列は、S S 9 と S S 6 であるから、

SR5. SS [0] : 含まない, SR5. SS [1] : 含まない, SR5. SS [2] : 含む, SR5. SS [3] : 含む
と表現することができる。

【0051】

この系列をサイクリックに用いるとすると、仮想境界点列セグメントを生成すべき箇所は、SR5. SS [3] = SS9からSR5. SS [2] = SS6であるから、結局は「含まない」断面系列の連続区間を挟む断面系列に合致していることがわかる。

【0052】

上記した例では、SR5. SS [0] ~ SR5. SS [1] が「含まない」連続区間であり、この区間を挟むものが、一方はSR5. SS [3] = SS9であり、他方はSR5. SS [2] = SS6である。

【0053】

これを一般化すると、SS番号をそのまま用いるのではなく、「SRから参照されるSSを順に並べてサイクリックに用いる場合の、含まれていないSS連続区間の前にあるSSが開始側であり、後にあるSSが終了側である」というルールになる。

【0054】

さらに、「開始側に関しては、SRに最も近い断面の内、SRに向かって右側の（最後の）境界点を開始境界点とし、終了側に関しては、SRに最も近い断面の内、SRに向かって左側の（最初の）境界点を終了境界点とする」というルールを追加する。

【0055】

これらが、1つのSRと開始側のSSと終了側のSSとが指定された場合に、接続パターンを決め、開始境界点と終了境界点とを特定するためのルールである。開始側のSSと終了側のSSとを入れ替えると、他方の仮想境界点列セグメントが選ばれることになる。

【0056】

なお、SRに接続するSSが3つ以上ある場合には、いったんペアの組み合わせ

せに置き換えてそれぞれに上記したルールを適用することにより、正しく処理することができる。また、図 2 4 の $SR5, SS[3]$ と $SR5, SS[0]$ がこの順序で指定された場合には、その間に抜けている SS がないことから、実境界点列をそのまま使えばよい。また、 $SR5, SS[i]$ と $SR5, SS[i]$ とが指定された場合、つまり全く同一の SS が指定された場合、この指定は端点として処理することに相当するが、上記した処理で例外なく処理することができる。

【 0 0 5 7 】

次に、上記したように特定された 2 つの境界点の間に、どのように仮想境界点列を生成するかについて説明する。

【 0 0 5 8 】

最も簡単な方法は、2 つの境界点を直線状に結ぶときに通過する座標値を求めて、それを座標値とする仮想的な境界点を生成する方法である。この方法は、曲線的な滑らかさが失われ、接続箇所で不要な角が生成されるので、認識精度を低下させる恐れがある。

【 0 0 5 9 】

そこで、本発明では、2 つの境界点の間を直線状に結ぶのではなく、 CG や フォント生成などの分野で用いられる曲線発生方法を用いる。具体的には、2 点の座標値 $P0$ 、 $P1$ およびその接線方向 $P0'$ 、 $P1'$ の推定値が得られるので、 $Ferguson$ の方法等を用いることができる。 $Ferguson$ の方法とは、上記 4 つのベクトルを 3 次多項式で結合したパラメトリックな方法であり、曲線発生方法として知られている。本発明は、 $Ferguson$ の方法に限定されず、他の曲線発生方法を用いてもよい。

【 0 0 6 0 】

本発明では、上記したように曲線発生方法を用いて仮想境界点列を生成することができるので、認識精度の向上に効果がある。

【 0 0 6 1 】

なお、線幅が非常に小さい場合には、仮想境界点列セグメントの長さが短くなることから、補間の精度が文字認識の精度にほとんど影響しないことがある。ま

た、両端の境界点がきわめて近い場合にも、同様である。従って、このような場合には、あえて仮想境界点を生成することなく、両端の境界点を論理的に直結すればよい。具体的には、境界点の接続情報のみを変更すればよい。これにより、処理量が削減される。

【 0 0 6 2 】

以上説明したように、本発明に係る仮想境界点列で重要な点は、「仮想境界点列は、画像平面上にプロットされるのではなく、あくまでも断面系列グラフという特徴表現上で実現される」ことにある。

【 0 0 6 3 】

(タグの表現形式)

本発明の処理においては、タグと呼ばれる論理的な構造要素が使用される。図 2 5 は、タグの表現形式を示す。タグ構造体 7 0 は、ステータス 7 1 (有効/無効、統合許可有り/なし、およびタグの種類を示すフラグ)、子の数 7 2 (次の 2 つの配列の有効数)、子の番号列 7 3、子の種類列 7 4 (この番号列と種類列は互いに対応する配列)、外接座標値 7 5 (X s、Y s)、7 6 (X e、Y e)、リンク先の数 7 7 (次の 2 つの配列の有効数)、リンク先のタグ番号の列 7 8、リンク先の評価値の列 7 9 (タグ番号の列と評価値の列は互いに対応する配列)、タグの評価値 8 0、認識結果番号 8 1、その他のワーク変数 8 2 (特異領域の接続パターンを示す変数、対応する境界点数、ワークフラグなどの一時的に使用される変数群) からなる。

【 0 0 6 4 】

上記したように、タグ構造体には、2 種類の配列ペアがある。前者の配列は、子を管理するものであり、これによって他のタグあるいは断面系列グラフの構造要素を統一的に管理することができる。概念的には、タグと断面系列グラフの構造要素との縦のつながりを表す。

【 0 0 6 5 】

後者の配列は、リンク先タグを管理するものであり、概念的には文字列方向のつながりを表す。タグ構造体は、前述した断面系列グラフの場合と同様に、メモリ上の連続領域として管理される。

【 0 0 6 6 】

タグは、子となるタグの内容に合わせて幾つかの種類に分けられる。要素タグとは、白背景ではなく黒成分の構造要素を子とするタグである。本発明では、断面系列グラフとの組み合わせで説明するので、以下の3種類とするが、一般にはこの限りではない。以下の説明は、図2を参照。

【 0 0 6 7 】

要素タグの第1はSSタグであり、断面系列を子とするタグである。要素タグの第2はSRタグであり、特異領域を子とするタグである。要素タグの第3はBCCタグであり、境界点列を子とするタグである。BCCは黒連結成分 (Black Connected Component) の略である。ただし、後述するように、実際に黒連結成分を求めるわけではない。また、境界点列の一部が仮想境界点列である場合は、仮想境界点列を子とするタグをVCタグと表記する。VCは、仮想境界点列 (Virtual Contour) の略である。

【 0 0 6 8 】

なお、図25のタグ構造体では子の種類を個別に設定できるようになっているため、1つのタグが異なる種類の子をもつことができるが、説明が複雑化することを避けるために、そのようなタグを考慮せず、上記した名称を用いる。

【 0 0 6 9 】

VSタグとは、子をもたないタグであり、図25の外接座標値を利用して空白領域を表現するのに用いられる。VSは仮想空白 (Virtual Space) の略である。文字列の始端および終端に対応するタグを、それぞれRタグおよびEタグと表記する。それぞれ、RootとEndの略である。

【 0 0 7 0 】

VCCタグとは、要素タグ (SSタグ、SRタグ、BCCタグ) またはVSタグを子とするタグである。VCCは仮想連結成分 (Virtual Connected Component) の略である。VCCタグは、切り出し方に関して1つの文字候補を表し、文字認識の処理対象となるものである。

【 0 0 7 1 】

パスタグとは、VCCタグとVSタグの総称であり、パス生成とパス選択の対

象となる。

【0072】

リンクとは、既に述べたように、文字列方向におけるタグ間の接続関係を表すものであり、主としてパスタグで用いられる。

【0073】

パスとは、リンクを介してパスタグを順に並べたものであり、Rタグに始まりEタグに終わる。文字列に対する1つの解釈候補を表す。

【0074】

概念的には、タグもまた階層的な記述を実現するものである。下位層として要素タグおよびVSタグを、上位層としてパスタグ（VCCタグとVSタグ）を考えることができる。VSタグは両方にまたがっている。これは、空白については、それが文字間空白である場合と文字内空白である場合との両方があり、両者に対応するためである。

【0075】

要素タグは、さらに2つの階層に分けて考えることができる。上位層がBCCタグであり、下位層がSSタグ、SRタグである。

【0076】

そして、要素タグより下位の層を構成するのが断面系列グラフである。ただし、断面系列グラフでは、BCCが下位層で、SS、SRが上位層にあたるが、要素タグとしては、SSタグ、SRタグが下位層で、BCCタグが上位層になっている。これは、複数の境界点列を集めて黒連結成分に相当するものをBCCタグとしていて、より粗い表現と見なせるからである。

【0077】

タグには、2つの役割がある。その1つは、文字要素（構造要素）と文字認識手段との間を架橋するという役割である。これは、VCCタグから要素タグを介して構造要素にアクセスできるという形で実現される。これが「特徴要素の組み合わせ」を実現するうえで、有効に作用する。また、要素タグを導入することによって、構造要素の種類の違いを吸収することができる。

【0078】

他の1つは、パス選択を中心とする文字列認識の操作対象としての役割である。この2つの役割を兼ね備えているところに、タグの特徴がある。このように処理対象の表現形式がタグとして共通化されているので、様々な処理を共有できるようになる。したがって、実現手段の簡易化を図ることができる。

【 0 0 7 9 】

こうしたタグを中心として全体を制御するためのタグ制御構造体を図26に示す。図26において、処理モード変数群は、ユーザに指定された文字種や機能（一文字認識／文字列認識機能）、および層優先度を示すための変数群である。層優先度とは、断面系列グラフをどの層まで求めるか、要素タグの何れの層を優先するか、に係わる情報である。

【 0 0 8 0 】

続いて、断面系列グラフの構成要素および仮想境界点配列へのポインタ群、およびタグ配列へのポインタが並ぶ。

【 0 0 8 1 】

図29は、タグのソート結果配列を示す。この配列は、タグを直接に並び替えるのではなく、タグに番号を付け、その番号を並び替えた結果を示す。この配列の上限は一定であるが、その配列の有効数は処理の過程で変化する。

【 0 0 8 2 】

さらに、パスを表現する配列、認識結果配列へのポインタ、文字列の幾何的情報等の変数群が続く。タグ制御構造体の最後に、特徴あるタグの番号群として配列がある。図28は、タグ配列の使用を説明する図である。タグ配列は、先頭から順に使用されるので、処理の過程でその有効数が次第に増えていくことになる。しかしながら、図28に示すように、特徴的なタグの番号を覚えておくことができ、これによって各ステップで処理対象となるタグの範囲を限定することができる。

【 0 0 8 3 】

例えば、

要素タグは、先頭から最初のRタグの直前まで

VSタグは、RタグからEタグまで

下位層のタグは、先頭からEタグまで（要素タグ+VSタグなので）
 VCCタグは、VCCタグの最初からVCタグの直前まで
 上位層タグは、RタグからVCタグの直前まで（VSタグ+VCCタグ）
 となる。

【0084】

これをソート結果配列と併用すれば、タグの入れ替えを回避したまま処理範囲を限定することが可能となる。

【0085】

以下、本発明の実施例を説明する。図1は、本発明の実施例の概要を説明する処理フローチャートである。また、図2は、本発明に係るタグの階層を説明する図である。図1、2を参照して本発明の処理の概要を説明する。

【0086】

画像データを入力し（ステップ101）、画像データから断面系列グラフを抽出する（ステップ102）。次いで、断面系列グラフを管理するための要素タグを生成する（ステップ103）。要素タグの上位層であるBCCタグは、断面系列グラフの下位層を管理し、要素タグの下位層であるSSタグとSRタグは、断面系列グラフの上位層を管理する。

【0087】

続いて、断面系列の特異領域に対して仮想境界点列を生成し（ステップ104）、要素タグを組み合わせることにより文字候補を生成する（ステップ105）。文字候補はVCCタグで管理される。

【0088】

次いで、文字候補に必要な仮想境界点列を補ってから一文字を認識し（ステップ106）、一文字認識された文字候補間にリンクを生成する（ステップ107）。リンクをたどってパスを生成し、最適なパスを選択して（ステップ108）、文字列の認識結果を出力する（ステップ109）。

【0089】

図3は、本発明の実施例の構成を示す。図3において、1は画像データを入力する画像入力部、2は画像データから断面系列グラフを抽出する断面系列処理部

、3はタグを生成し管理するタグ処理部、4は仮想境界点を生成する仮想境界点処理部、5は各種構造体などを格納するメモリ、6はリンクおよびパスを生成するリンク／パス生成部、7は文字認識部、8は全体を制御する制御部である。なお、入力される画像データは2値でも多値でもよいが、多値の場合は、所定の閾値で画像の白／黒を判定しながら処理を行う。

【0090】

図4、5は、本発明の処理に係る処理全体のフローチャートを示す。ステップ201の準備処理では、以下の処理に必要な情報を取得する。すなわち、処理対象となる文字列画像を画像入力部1から入力する。処理対象となる文字列画像の例を図30に示す。またユーザ指定情報を入力する。このユーザ指定情報は、文字種や機能（一文字認識または文字列認識を示す情報）などを指定する情報である。

【0091】

ステップ202において、制御部8は、取得された情報を、メモリ5内のタグ制御構造体（図26）の処理モードに設定し、その内容に基づいて層優先度を設定する。

【0092】

ステップ203において、設定された処理モードに応じて、断面系列処理部2は、断面系列グラフを抽出し、メモリ5内に図16～図20の形式で格納する。図31には、断面系列グラフの下位層（境界点列）と、断面系列グラフの上位層（断面系列SSと特異領域SR）を示す。図31は、タグ構造体をモデル化した図である。

【0093】

なお、BCCタグが優先であれば、断面系列グラフにおける上位層（SS・SR）は求める必要がないので、処理を省略できる。

【0094】

ステップ204において、タグ処理部3は、断面系列グラフの下位層にある境界点列に基づいて、BCCタグを求める。この処理について、図31の境界点列（C0～C2）を例にして説明する。

【 0 0 9 5 】

まず、外側境界点列 C 0、C 1 に対して、対応するタグをそれぞれに作り、これらを B C C タグ 0、B C C タグ 1 としてメモリ 5 に登録する（図 2 5）。B C C タグの子として、対応する外側境界点列 C 0、C 1 がメモリ 5 に登録される（図 1 7）。このとき、境界点数が所定の閾値以下である外側境界点列は、ノイズの可能性が高いので、B C C タグ作成の対象から除く。

【 0 0 9 6 】

次に、内側境界点列 C 2 に対しては、新たに B C C タグを作成せずに、上記した B C C タグから対応するものを求めて、B C C タグ 0 の子として登録する。具体的には、断面系列グラフ上で、境界点列の対応関係を求めたうえで、その結果をタグに反映する。つまり、境界点が画像を走査する順に抽出されていることを前提にすると、内側境界点列から任意の境界点を一つ取り出し、境界点情報を登録の逆順に取り出し、最初に出現する境界点（ただし外側境界点列に属するもの）を見つければよい。この外側境界点に対応する外側境界点列を子としてもつものが、求めたい B C C タグである。

【 0 0 9 7 】

上記した B C C タグの作成処理で重要な点は、黒連結成分そのものを求めているのではなく、必要に応じて黒連結成分の境界点列と境界点を取り出すための構造化情報を作成している点である。

【 0 0 9 8 】

ステップ 2 0 5 において、タグ処理部 3 は、得られた B C C タグの数を調べる。タグ数が 0 個であれば、制御部 8 は、入力された文字画像は空白であると判断して、図 5 のステップ 2 1 7 に進み、空白コードを設定して、処理を終了する。タグ数が 0 でなければ、ステップ 2 0 6 に進む。

【 0 0 9 9 】

ステップ 2 0 6 ～ステップ 2 1 6 の処理において、文字列認識が実行される。ステップ 2 0 6 において、タグ処理部 3 は、断面系列グラフの上位層（S S と S R）を基に、それぞれに対応するタグを求める。すなわち、図 3 1 の特異領域 S R 1 に対応して S R タグ 1 を作成し、断面系列 S S 0 に対応して S S タグ 0 を作

成し、以下同様にして、SR0に対応してSRタグ0を、SS2に対応してSSタグ2を、SR2に対応してSRタグ2を、SS1に対応してSSタグ1を、SR3に対応してSRタグ3を作成する。

【0100】

なお、ステップ203において、処理モードの層優先度に従って断面系列グラフの上位層が求められていない場合には、ここで作成されるタグは0個となるが、要素タグとしては既にステップ204で求めたBCCタグがあるので、以下の処理を継続することが可能である。

【0101】

ステップ206の処理は、ステップ204と対応関係にある。この時点で、図31に示されるように、必要なすべての要素タグが求められる。図31の構造は、図26のタグを制御する構造体に格納されている。

【0102】

ただし、BCCタグとSSタグ、SRタグは、それらが表現している文字要素に重複があるので、後述するステップ211において取捨選択を行うことにより、重複を解消する。

【0103】

ステップ207において、制御部8は、要素タグに基づいて文字列全体に関する特徴（例えば文字の高さなど）を抽出する。抽出された特徴は、図26の構造体中の、文字列の幾何的情報等変数群の内容として設定される。これらは、後述するように、文字らしさを評価する処理（ステップ213）などで用いられる。また、ステップ207において、タグ処理部3は、要素タグをソートする。図31の要素タグ（タグ番号）をソートした結果を図32に示す。タグ番号のソートは、図31に示す断面系列グラフの上位層（SSとSR）の左からの配置順通りに並べ替えたものである。BCCタグの番号（0、1）も同様に左からの配置順通りに並べ、かつSSとSRタグの番号の間に挿入する。その挿入位置は、境界点列C0、C2の外接矩形の重心位置（文字列横方向の位置）、C1の外接矩形の重心位置（文字列横方向の位置）を考慮して決める。

【0104】

ステップ 2 0 7 の時点でのソート結果は、図 2 9 に示す通りであり、矢印がソート対象の範囲を示す。

【 0 1 0 5 】

ステップ 2 0 8 において、タグ処理部 3 は、B C C タグの組み合わせについて、画像平面上の位置に換算した場合の隣接性を調べ、互いに隣り合う B C C タグ・ペアの間隙（文字列方向）に対応する V S タグを求める。図 3 3 は、断面系列グラフの下位層（境界点列）と共にタグ番号の配列を示す。また、図 3 3 には V S タグ 0 も示されている。

【 0 1 0 6 】

V S タグは、前述したように（仮想）空白を表現するためのものである。B C C タグは、実質的には黒連結成分に相当するものであるから、画像平面上で隣接関係にある B C C タグ間（図 3 3 のタグ番号 0 と 1）には、必ず間隙がある。ただし、文字列方向と垂直な方向に完全な空白領域がある場合もあれば（図 3 3 の例）、そうでない場合もある。そうでない場合とは、斜めの空白領域であったり、入り組んだ空白領域であったりする場合である。このような場合でも、仮想的な空白領域があると見なして、V S タグとして登録する。上記した斜めの空白領域や入り組んだ空白領域は、その座標値を調べることにより、文字切り出し枠が重なった場合（これを負の空白という）であることが分かる。図 3 4 は、負の空白例を示す。

【 0 1 0 7 】

また、タグ処理部 3 は、文字列方向に十分な大きさをもつ V S タグを、文字内空白ではないと判断して、図 2 5 に示すタグ構造体のステータスに「統合を許容しない」と設定する。このステータスは、後述するステップ 2 1 3 の処理に反映され、V C C タグ数の増大を抑制するのに寄与する。

【 0 1 0 8 】

また、タグ処理部 3 は、文字列の両端には仮想的な空白があると判断して、始端に相当する空白を R タグとし、終端に相当する空白を E タグとする。既に述べたように R タグも E タグも V S タグの一種である。図 3 3 の要素タグには、R タグ、V S タグ、E タグが追加されている。

【 0 1 0 9 】

さらに、ステップ 2 0 8 では、要素タグに V S タグを加えてソートする。そのソート結果を図 3 5 に示す。また、この時点でのソート結果は、図 2 9 に示すように、E タグまでの矢印がソート対象の範囲を示す。

【 0 1 1 0 】

以上の処理により、要素タグ（B C C タグ、S S タグ、S R タグ）と V S タグが求められたことになる。ただし、要素タグはパスタグではないので、そのままパスの構成要素（つまり一文字に相当すると見なされるもの）とはならない。パスの構成要素となる V C C タグは、要素タグの様々な組み合わせを 1 つのタグで表現するものであり、それらはステップ 2 1 3 で求められる。

【 0 1 1 1 】

ステップ 2 0 9（図 5）において、タグ処理部 3 は、要素タグを対象にして、予約処理を行う。この予約処理には以下の 4 つの役割がある。

【 0 1 1 2 】

第 1 は、ステップ 2 1 0 以下の通常の処理では例外として処理すべき文字、具体的には句読点のように特殊な大きさの文字を、この時点で事前に認識することにある。その理由は、句読点などの特殊文字は、周囲の文字との配置状況に応じて決められるものであり、配置によらずに形状などに基づく一文字認識だけでは精度よく判断することが困難なためである。この時点で適切に処理することにより、後処理において例えば大きさを評価するなどの特別な処理を行わなくて済む。

【 0 1 1 3 】

第 2 は、バックトラック型アプローチと、パス選択型（または仮説検証型）アプローチを融合することにある。予約処理とステップ 2 1 1 の「要素タグの取捨選択」の組み合わせによってバックトラック型となり、さらにステップ 2 1 5 とそれ以後の処理の組み合わせによって仮説検証型（またはパス選択型）となる。より具体的には、上位層（B C C タグ）に対して予約を行い、予約されない場合に限って対応する下位層（S S ・ S R タグ）を残し、残った要素タグについて、ステップ 2 1 3 でその組み合わせを作成する。

【0114】

第3は、特殊な文字認識方法を効果的に導入することにある。この時点で、高品質文字を早めに確定することができれば、VCCタグ数やパス数を抑えることができる。

【0115】

第4は、適切なノイズ対策である。文字列画像全体に、一様にノイズ除去を適用する場合、例えばカタカナの「ウシソツホミン」などの小さな文字要素を消失することにもなるため、あまり強いノイズ除去を適用することができない。そこで、隣接関係を考慮した上で、ノイズ除去を実現するために予約処理を利用する。この時点で一定の大きさを満たさないものに、「ノイズの可能性がある」という予約をする。これはステータスに反映される。その結果をステップ213でVCCタグに継承し、ステップ214、215においてパスとしての評価に反映させることで、隣接関係を考慮した上でのノイズ除去が実現される。

【0116】

図6は、要素タグの予約処理の詳細な処理フローチャートを示す。ステップ301では、処理モードを解釈して、予約処理で用いられる基本条件を設定する。すなわち、文字種条件を基に予約処理を制御する。例えば、微小記号を表す文字種が指定されているか否か、複数の文字認識方法の内、使用する認識方法などの情報である。予約の必要がなければ（ステップ302でYES）、この処理を終了する。予約が必要である場合（ステップ302でNO）には、以下の処理を行う。

【0117】

ステップ303とステップ304は、要素タグを順に取り出す繰り返し処理である。処理範囲は、図28において、先頭からRタグの直前までである。

【0118】

ステップ305では、取り出した要素タグが処理対象であるか否かを調べる。ここでは、BCCタグか否かを調べ、BCCタグならばステップ306に進み、そうでなければステップ303に戻る。

【0119】

ステップ 3 0 6 では、一時的な V C C タグを準備・初期化し、ステップ 3 0 7 では取り出した B C C タグを、一時的な V C C タグの子として登録する。

【 0 1 2 0 】

ステップ 3 0 8 では、一時的な V C C タグを対象として後述する一文字認識を行う。

【 0 1 2 1 】

ステップ 3 0 9 では、文字認識結果を B C C タグに反映する。本来は文字認識結果は V C C タグに反映されるが、タグ構造体としては共通の形式を用いているので、B C C タグに反映することが可能である。

【 0 1 2 2 】

上記した文字認識結果に応じて、処理内容が以下の 3 通りの何れかとなる。すなわち、

(a) 認識結果を捨て、B C C タグに反映しない場合

(b) 認識結果を残し、B C C タグに反映する場合

(c) 認識結果を修正し、B C C タグに反映する場合

である。

【 0 1 2 3 】

図 3 6 (a) は、認識結果を捨て、B C C タグ 0 に反映しない例を示し、図 3 6 (b) は、認識結果 (" 2 ") を残し、B C C タグ 1 に反映する例を示す。

【 0 1 2 4 】

認識結果が拒否あるいはそれに類似する（確信度などによる判定を含む）場合などが (a) に相当し、この場合は、図 2 7 の認識結果構造体のステータスに再認識の必要性が指定されている。この他にも、位置、大きさが特殊文字であって、認識結果の第 1 位候補が特殊文字ではない、多くの場合がこれに相当する。

【 0 1 2 5 】

一方、タグの位置情報を、ステップ 3 0 1 で定めた条件を参照することにより、特殊文字であると判定した場合には、文字認識結果の第 1 位候補の内容を調べる。第 1 位候補が位置情報による推定結果と一致する特殊文字である場合は (b) に相当する。

【0126】

特殊文字ではないが、あらかじめ定められた特定の文字コードであれば、認識結果の第1位候補に適切な特殊記号の文字コードを挿入したうえで、BCCタグに反映する。これは(c)に相当する。これにより、例えば、低い位置にある小さなタグが「ノ」と判定されたものに対して、第1位候補に「,」を挿入するという処理が実現できる。

【0127】

このような候補挿入ルールは、一文字認識手段の特性に応じてあらかじめ定められている。また、他のルールとして、一文字認識手段において、どのような方法を組み合わせて文字認識したかによって、結果を無視する場合と、残して反映する場合があらかじめ定められている。

【0128】

ステップ309では、上記したような文字認識結果に応じた処理を行う。つまり、タグについて、位置や大きさなどを含む配置条件と文字種条件と文字認識結果(どのような手段で認識されたかを含む)を調べて、上記した(a)~(c)の処理を選択し、実行する。(b)~(c)の場合には、さらにタグの統合許可の有無も設定され、これらは図25のタグ構造体70のステータス71に設定される。

【0129】

つまり、タグは、予約されない場合(a)と、予約される場合(b)、(c)があり、これらは接触文字を分離するか否かを制御することに係わる。

【0130】

これに対して、予約される場合には、統合許可の有無に応じて、分離文字を統合するか否かを制御する。

【0131】

この時点でBCCタグに反映された情報は、図5のステップ213を介して図8のステップ512によりVCCタグへと継承される。

【0132】

すべての要素タグに対する予約処理が終了したならば(ステップ304でYE

S)、ステップ310において、処理モードの文字種条件から特殊記号を除く。

【0133】

図5に戻り、ステップ210において、仮想境界点処理部4は、SR（特異領域）を解析し、仮想境界点列を生成する。図7は、SR（特異領域）解析による仮想境界点列生成処理の詳細フローチャートを示す。

【0134】

ステップ401では、図26に示すタグ制御構造体の処理モードに設定されている層優先度を調べて、BCCタグが優先であれば、以下の処理をスキップする。

【0135】

ステップ402～ステップ404は、要素タグの中から、有効なSRタグを取り出すループ処理である。このときの処理対象は、図28のRタグの直前までとなる。

【0136】

ステップ405では、SRタグからSR（特異領域）を取り出す。ステップ406では、SRに接続するすべてのSSペアを求める。図24において、4つのSSがSRに接続しているが、この場合にはSSペアの数は、その2乗の16となる。

【0137】

ステップ407～ステップ408はSSペアを順に取り出すループ処理である。ステップ409では、図23のように特異領域の接続パターンを求める。ステップ410では、接続パターンに応じて、狭義の仮想境界点列の必要性を判定する。このとき、実境界点で代用できるものは不要とされる。

【0138】

また、接続パターンが、図24において例えばSS3を文字の一部とし、SS6とSS7とSS9を文字の一部とするように、SS3と、SS6、7との間で上下に切る接続パターンであるとき、文字列方向が横方向であれば、SS3が右側の文字の一部、SS6とSS7とSS9が左側の文字の一部となることはあり得ず、従って、上記したような接続パターンに対しては狭義の仮想境界点列を生

成しない。このように、文字列方向からみて不自然な断面系列の間には仮想境界点列を生成しない。

【0139】

ステップ411では、ステップ409で求めた接続パターンに基づいて仮想境界点列セグメントを求める。ここでは、既に述べた方法を用いて、適切な仮想境界点列セグメントを求める。

【0140】

ステップ412では、仮想境界点列セグメントと接続情報を、図22の形式で仮想境界点列を登録する。

【0141】

図5に戻り、ステップ211において、タグ処理部3は、要素タグの取捨選択を行う。取捨選択が必要となるのは、BCCタグとSS/SRタグが正しく得られ、処理モードにおいてSS/SRタグ優先が設定されている場合に限られる。この場合には、BCCタグとSS/SRタグの間に重複があり、その解消が必要となる。

【0142】

まず、層優先度を決定する。最も簡単な決定方法は、ユーザが層優先度を指定する方法である。これは、ユーザの使用環境と目的に応じた、「認識精度を優先する」あるいは「処理速度を優先する」などの処理要求となる。

【0143】

他の決定方法は、予め文字種の組み合わせに応じて、層優先度を定義しておき、ユーザに指定された文字種情報に応じて自動的に層優先度を選択する方法である。この方法は、ユーザの負担を軽減できる。

【0144】

しかし、上記した2つの方法では、何れか一方の層のみが用いられるので、処理が簡単であるものの、処理精度と処理量のバランスを図ることが難しい。

【0145】

そこで、本発明では、処理精度と処理量のバランスを図るために、層優先度を自動化する方法を考える。その1つの方法として、ステップ209の予約処理の

結果を用いる方法を、以下説明する。

【0146】

ステップ209の予約処理の結果に基づいて、要素タグを順に取り出して、それがSSタグまたはSRタグであれば、これと重複するBCCタグを取り出す。このBCCタグが、ステップ209において予約済みであれば、BCCタグを残して、SS/SRタグのステータスを除去状態とする。そうでなければ、BCCタグのステータスを除去状態とし、SS/SRタグをそのまま残す。このような処理によって、ステップ211が終了した時点では、残されている要素タグには重複がない。

【0147】

図37(a)、(b)は、要素タグの取捨選択を説明する図である。(a)は、ステップ209でBCCタグが破棄されているので、SSタグ0、SSタグ2、SRタグ0、SRタグ1を残す。(b)は、ステップ209でBCCタグが予約済みであるので、SSタグ1、SRタグ2、SRタグ3を破棄する。また、図38は、要素タグの取捨選択された結果を示す。

【0148】

ステップ212は、残された要素タグの初期統合を行う。これは、同じ種類の要素タグの組み合わせに関して、それらの文字列方向の重なりが大きければ、両者を統合する処理である。例えば、文字列方向が横方向である場合には、偏や旁などはそれぞれに統合されやすい。統合は、一方の子タグを他方の子タグにコピーして子の数を更新し、前者のステータスを除去状態にすることにより実現される。

【0149】

図39は、要素タグの初期統合の例を示す。境界点列C5とC6が文字列方向に重なっているので、境界点列C5とC6を子とするそれぞれのBCCタグを統合する。

【0150】

ただし、ステップ209の予約処理の結果として、「統合を許容しない」と設定されている要素タグは、ステップ212の統合処理を行わない。また、ステッ

プ 2 1 2 では、要素タグと V S タグを合わせて文字列方向に沿って、あらためてソートする（図 2 9）。

【 0 1 5 1 】

ステップ 2 1 3 において、タグ処理部 3 は、要素タグを組み合わせて V C C タグを生成する。V C C タグとは、文字切り出し処理から見た場合の「一文字になりうる候補」を意味し、要素タグの 1 つ以上の組み合わせで表現される。つまり、要素タグおよび V S タグが V C C タグの子となる。

【 0 1 5 2 】

図 8、9 は、ステップ 2 1 3 の詳細の処理フローチャートである。図 8、9 の処理は、文字列方向として横方向（左から右へ）の場合を仮定しているが、縦方向であっても同様に処理できる。図 4 0 は、V C C タグの生成過程を示す図である。

【 0 1 5 3 】

ステップ 5 0 1 では、以下の処理で一時的に用いるための V C C タグを準備する。ステップ 5 0 2 ～ステップ 5 0 4 が処理対象となるタグ（要素タグまたは V S タグ）を順に取り出して、左端とするループ処理であり、同様に、ステップ 5 0 6 ～ステップ 5 0 8 が処理対象となるタグを順に取り出して、右端とするループ処理である。また、右端を取り出すループ処理は、左端と同じ要素タグから始めるものとする。

【 0 1 5 4 】

上記した処理により、ステップ 2 1 2 のソート時に連続して並ぶタグのすべての組み合わせが作成される。このように、連続性を条件としてタグを組み合わせているので、タグの全ての組み合わせを求める場合に比べて、V C C タグ数が削減される。

【 0 1 5 5 】

なお、V S タグは、要素タグの両端のいずれにも用いずに、要素タグに挟まれる場合のみを用いられる（図 4 0 の V C C タグ 9、V C C タグ 1 1）。また、ステップ 5 0 8 において、取り出したタグが V S タグであり、かつそのステータスが「統合を許容しない」場合には、ステップ 5 0 2 に戻る。

【0156】

ステップ509の処理時点では、上記のループ処理で取り出された右端の要素タグを一時的なVCCタグの子タグとして登録する。ステップ510では、所定の条件（例えば文字の幾何的な条件）を参照して、VCCタグを残すか否かを判定する。所定の条件（例えば、右端がVSタグである）に一致する場合には、ステップ510で対象外であると判定されて、ステップ506に戻る。なお、上記した所定の条件として、文字列方向の大きさなども追加できる。

【0157】

ステップ511では、正式なVCCタグを一つ取り出して、一時的なVCCタグの内容をコピー（登録）する。つまり、図40に示すように、要素タグにVCCタグを追加する。

【0158】

ステップ512では、いま追加した子タグ、すなわち右端の要素タグが予約済みであるか否かを調べ、予約済みである場合には、一文字認識結果に関する内容を継承する。つまり、VCCタグの子であるBCCタグが、例えば図36の（b）のように認識結果を保持している（予約済み）場合には、その認識結果をVCCに渡す。さらに、予約の結果としてステータスが「統合を許容しない」場合には、これ以上右端を先に進めることができないので、ステップ502に戻って、次の左端を取り出す。「統合を許容する」場合や、予約済みでない場合には、ステップ506に戻って、右端を先に進める。

【0159】

左端、右端ともにすべての要素タグの取り出しが終了すると（ステップ503でYES）、図9のステップ514に移る。

【0160】

ステップ514、ステップ515は、VCCタグを順に取り出すループ処理である。ステップ516では、後述する一文字認識を行う。ステップ517では、その結果をVCCタグに反映させる。つまり、VCC（図25）の認識結果番号81に認識結果を設定する。すべてのVCCタグについて、一文字認識が終了したならば、図5のステップ214に移る。

【 0 1 6 1 】

図 5 に戻り、ステップ 2 1 4 において、リンク生成／パス生成部 6 は、パスタグ（VCC タグおよび VS タグ）を対象として、リンクを生成する。これはパスを生成するための、局所的な隣接関係を定めるためである。図 4 1 は、生成されたリンクを示す図である。

【 0 1 6 2 】

ステップ 2 1 5 では、パスタグとそのリンクを用いて、図 4 2 に示すようにパス（例えば R→V0→V4→V7→V11→E）を生成し（図 4 2 では、VCC タグ 0 を V0、VCC タグ 1 を V1 と表す）、最も最適なパスを選択する。図 4 3 は、選択されたパスを示す。上記したパスとリンクは木構造を構成しているので、例えば DP マッチングなどを適用して最適なパスを選択する。

【 0 1 6 3 】

ステップ 2 1 6 では、ステップ 2 1 5 で選ばれた最適パスを対象として、特定の文字の再認識処理を実行する。その処理内容は、一文字認識の処理を、文字列へと拡張したものに相当し、処理の大部分は共有されている。

【 0 1 6 4 】

図 1 0 は、ステップ 2 1 6 の詳細な処理フローチャートである。ステップ 6 0 1 ～ステップ 6 0 3 は、最適パスから VCC タグを順に取り出すループ処理である。

【 0 1 6 5 】

ステップ 6 0 3 では、VCC タグからその認識結果を取り出して、以下の処理の対象となるか否かを調べる。本発明では、認識結果構造体（図 2 7）のステータスに再認識の必要性が示されている場合には、ステップ 6 0 4 以下の処理を実行する。

【 0 1 6 6 】

ステップ 6 0 4 では、VCC タグから必要な画像のみを再構成する。具体的には、フォント表示などでよく用いられている方法で実現できる。VCC タグから要素タグを介して境界点を取り出し、初期化された画像データ上にこれらをプロットし、プロットされた黒画素間を黒画素で埋めれば、画像を再構成することが

できる。

【0167】

ステップ605では、文字画像に対して適応的なフィルタリングを行う。線が細い場合には境界点近傍に黒画素を増やし、太い場合には境界点近傍の黒画素を除去するようなフィルタを選択して適用する。

【0168】

ステップ606では、フィルタリングされた画像からあらためて断面系列グラフを抽出、記述する。これは、図4のステップ203と同一の処理である。

【0169】

ステップ607では、新しく得られた断面系列グラフに基づいてBCCタグを作成する。この処理は、図4のステップ204と同一の処理である。

【0170】

ステップ608では、1つの一時的なVCCタグを準備し、新しく得られたBCCタグを子として登録する。ステップ609では、BCCタグを子とするVCCタグを処理対象として後述する一文字認識を行う。

【0171】

ステップ610では、ステップ605～ステップ609まで処理におけるエラーの有無を調べる。エラーがなければ、ステップ611で、VCCタグの認識結果を置き換え、特定の文字の再認識が終了する。

【0172】

ステップ604～ステップ606では画像データへのアクセスが生じるが、実行される割合は数%にも満たない程度のものであるから、本発明の中心的な考え方である、断面系列グラフの上位層と下位層を同一表現形式のタグで管理するという重要性が失われることはない。逆にいえば、本発明のように特徴要素の組み合わせという考え方を採らない従来の方法の場合には、すべてのVCCタグについて、一文字認識における例えばステップ710とステップ711の間でステップ604と同一の処理を繰り返す必要がある。この場合には、画像データへのアクセスの割合が100%になることから、本発明により、相当な処理量の増大が抑制されることになる。

【 0 1 7 3 】

ステップ 2 1 7 では、以上の処理で得られた認識結果を設定する。以上で、全体の処理が終了する。

【 0 1 7 4 】

最後に、文字認識部 7 における一文字認識の処理を説明する。図 1 1 は、図 6 のステップ 3 0 8、図 9 のステップ 5 1 6、図 1 0 のステップ 6 0 9 における一文字認識の詳細な処理フローチャートである。

【 0 1 7 5 】

この処理の特徴は、要素タグの種類の違いを吸収しながら一文字認識を実際に行う点にある。また、基本的な考え方は、認識結果を境界点列表現に統一するところであり、そのために、必要ならば狭義の仮想境界点列を使って境界点列の一時的なつなぎ換えを行う。さらに、認識の手法としては、複数の文字認識手法が併用されているが、認識を行う際に使用される特徴は、画像から得られるものではなく、タグを介して断面系列グラフと仮想境界点列から得られるものであり、これによって、本発明の「特徴要素の組み合わせ」という考え方が実現されている。

【 0 1 7 6 】

ステップ 7 0 1 では、一時的な V C C タグ（空の V C C タグ）を準備して初期化する。

【 0 1 7 7 】

ステップ 7 0 2、ステップ 7 0 3 は繰り返し処理であり、処理対象の V C C タグからすべての要素タグを順に取り出す。ステップ 7 0 4 は、取り出した要素タグが B C C タグであるか否かを調べ、そうであれば、ステップ 7 0 7 において一時的な V C C タグに子として登録する。図 4 4 は、子が B C C タグである場合の例を示す。図 4 5 は、B C C タグが一時的な V C C タグの子として登録された図である。

【 0 1 7 8 】

図 4 5 の例では、B C C タグが 1 個であるので、ステップ 7 0 3 からステップ 7 1 0 に進む。B C C タグの子は境界点列 C 1 であるので、ステップ 7 1 0、7

11、712で何も処理せずに、ステップ713に進む。

【0179】

ステップ713では、処理モードに応じて、境界点列C1の文字認識を行って結果をBCCタグに保存する。ここでは、複数の文字認識方法（例えば、特許第2719202号に記載されたマッチング手法など）が用意されていて、処理モードに応じて、あらかじめ決められた認識方法を組み合わせて文字認識を実行し、その結果を統合してこのステップにおける認識結果とする。その詳細については本発明に直接関係しないので省略する。重要な点は、これらの文字認識方法に用いられる特徴である、境界点の濃淡傾き方向をベースとした特徴、境界点列のグラフ表現、および骨格のグラフ表現などが、断面系列グラフとの整合がよく、それらの特徴を画像平面から直接求めるのではなくて、タグを介して断面系列グラフから求められる点である。

【0180】

図45の例ではステップ714で何も処理せずに終了する。
ステップ702で取り出された要素タグがSS、SRタグである場合について、図46を例にして説明する。図46の例では、要素タグが3個（SRタグ1、SSタグ0、SRタグ0）あり、まず最初のSRタグ1を取り出す（ステップ706）。SRタグ1は端点であるので何もせず、次のSSタグ0を取り出す。SSタグであるので（ステップ705）、両端の4境界点を保存する（ステップ708）。すなわち、図47において、SR1側の断面の2つの境界点と、SR0側の断面の2つの境界点を保存する。4つの境界点は、境界点列の一時的なつなぎ代え（ステップ710）に必要な情報である。

【0181】

最後にSR0タグを取り出す（ステップ702）。ステップ706では、取り出した要素タグがSRタグであるか否かを調べ、そうであれば、ステップ709において、狭義の仮想境界点列を取得する。すなわち、まずSRタグを介して特異領域を求め、SSタグを介して求めた断面系列との組み合わせにより、特異領域の接続パターン（図23）を求める。図48は、SR解析による特異領域SR0の接続パターンの例を示す。ここでは、開始境界点を持つ断面番号と終了境界

点を持つ断面番号が共に 2 であることを示す。

【0182】

次に、図 4 のステップ 210 によって、予め図 22 の形式で求められている狭義の仮想境界点列を、図 23 の接続パターンをキーとして照合し、狭義の仮想境界点列を得る（図 47）。

【0183】

ステップ 706 で、NO となるのは、VS タグの場合である。
ステップ 710 では、ステップ 708 で求められ、一時的に保存されている境界点がある場合に限り、それらを出発点として、境界点構造体を接続順に取り出しつつ、必要時にはステップ 709 で求めた仮想境界点列への一時的な置き換えを実行する。すなわち、SS0 の SR0 側の断面を仮想境界点列で置き換える（図 48 の点線）。

【0184】

ステップ 711 では、4 個の境界点と、一時的に置き換えられた仮想境界点列をなぞって、広義の仮想境界点列を作成し（図 47）、広義の仮想境界点列 C3 を子とする BCC タグ（すなわち VC タグ）を作る（図 49）。このように、図 46 の断面系列と特異領域は、図 49 に示すように特異領域で分離されると共に境界点列に変換され、この境界点列に対してステップ 713 で認識処理が行われる。

【0185】

ステップ 712 では、ステップ 711 で作られた VC タグがある場合に限り、それらを一時的な VCC タグの子として登録する。

【0186】

ステップ 713 では、処理モードに応じて、文字認識を行って結果を VC タグに保存する。文字認識方法としては、複数の認識方法が用意され、処理モードに応じて、予め決められた認識方法を組み合わせて文字認識を実行し、その結果を統合し、このステップの認識結果とする。

【0187】

ステップ 714 では、ステップ 710 において一時的につなぎ換えられた仮想

境界点列があれば、それをつなぐ前の状態に戻す。以上の処理により、一文字認識が行われる。

【 0 1 8 8 】

接触文字列に対して、上記した処理を適用した場合の結果を図 5 0 に示す。ここでは、接触している文字 (a) が、断面系列グラフ (b) に基づいて生成した仮想境界点列 (c) を介して、互いに適切に抽出されていることがわかる。なお、(d) の処理結果は、特徴レベルで表現されているものを画像的に視覚化するために、境界点をプロットしたものである。

【 0 1 8 9 】

なお、本発明はソフトウェアによっても実現できる。図 5 1 は、本発明をソフトウェアによって実現する場合のシステム構成例を示す。認識対象となる画像は、スキャナなどから入力され、システムに取り込まれる。またはハードディスクに格納されている画像データ、あるいはネットワークを介してシステムに取り込まれた画像データが認識の処理対象となる。

【 0 1 9 0 】

システム構成において、CPU は、入力画像に対して前述した実施例の処理ステップや処理機能を実行することにより文字認識処理を行い、認識結果を表示装置に表示する。

【 0 1 9 1 】

上記した処理を実行するプログラムは、CD-ROM などの記録媒体に記録されていて、媒体に記録されたプログラムを CD-ROM 装置から読み込み、システムにインストールすることによって実行され、上記実施例で説明した処理機能が実現される。また、上記したプログラムは、媒体の他、通信装置、ネットワークを介してサーバなどからダウンロードすることによっても提供される。

【 0 1 9 2 】

【発明の効果】

以上、説明したように、本発明によれば、断面系列グラフの特異領域に対して仮想境界点列を生成しているので、接触文字列を適切に切り出して高精度に認識することができる。

【 0 1 9 3 】

また、文字列が接触している特異領域に仮想境界点列が生成されるので、接触文字の分離が可能となる。

【 0 1 9 4 】

また、仮想境界点列が所定の曲線発生方法によって生成されるので、文字曲線が滑らかになり文字認識の認識精度が向上する。

【 0 1 9 5 】

また、階層が異なる画像特徴をタグで管理すると共に、タグの組み合わせによって文字候補を生成しているので、画像データへのアクセスが少なくなり、認識処理が高速化される。

【 0 1 9 6 】

また、タグを組み合わせる際に、複数の層にまたがるタグの重複を解消しているので、文字候補数を抑えることができると共に、処理量を削減できる。

【 0 1 9 7 】

さらに、タグの表現形式（論理構造）が同一であるので、処理の共用化が図られ、これにより装置構成が簡単化される。

【図面の簡単な説明】

【図 1】

本発明の実施例の概要を説明する処理フローチャートである。

【図 2】

本発明に係るタグの階層を説明する図である。

【図 3】

本発明の実施例の構成を示す図である。

【図 4】

本発明の処理に係る処理全体のフローチャートである。

【図 5】

図 4 の続きの処理フローチャートである。

【図 6】

要素タグの予約処理の詳細な処理フローチャートである。

【図 7】

S R（特異領域）解析による仮想境界点列生成処理の詳細フローチャートである。

【図 8】

VCCタグ生成の詳細な処理フローチャートである。

【図 9】

図 8 の続きの処理フローチャートである。

【図 1 0】

特定文字の再認識の詳細な処理フローチャートである。

【図 1 1】

一文字認識の詳細な処理フローチャートである。

【図 1 2】

線図形イメージデータ（原画像データ）と断面系列グラフ並びに骨格線（芯線）との関係を示す図である。

【図 1 3】

文字列画像の例を示す図である。

【図 1 4】

図 1 3 の画像に対する断面系列グラフの境界点と断面を示す図である。

【図 1 5】

図 1 3 の画像に対する断面系列グラフの断面系列と特異領域を示す図である。

【図 1 6】

境界点の表現形式を示す図である。

【図 1 7】

境界点列の表現形式を示す図である。

【図 1 8】

断面の表現形式を示す図である。

【図 1 9】

断面系列の表現形式を示す図である。

【図 2 0】

特異領域の表現形式を示す図である。

【図 2 1】

仮想境界点列セグメントを説明する図である。

【図 2 2】

仮想境界点列の表現形式を示す図である。

【図 2 3】

特異領域の接続パターンを説明する図である。

【図 2 4】

特異領域の接続の様子を説明する図である。

【図 2 5】

タグの表現形式を示す図である。

【図 2 6】

タグ制御構造体を示す図である。

【図 2 7】

認識結果の表現形式を示す図である。

【図 2 8】

タグ配列が使われる様子を示す図である。

【図 2 9】

タグ配列とソート結果配列の様子を示す図である。

【図 3 0】

処理対象となる文字列画像の例を示す図である。

【図 3 1】

タグ構造体をモデル化した図である。

【図 3 2】

タグ番号をソートした結果を示す図である。

【図 3 3】

断面系列グラフの下位層と追加される V S タグを示す図である。

【図 3 4】

負の空白例を示す図である。

【図35】

要素タグにVSタグを加えてソートした結果を示す図である。

【図36】

要素タグの予約処理を説明する図である。

【図37】

要素タグの取捨選択を説明する図である。

【図38】

要素タグの取捨選択された結果を示す図である。

【図39】

要素タグの初期統合の例を説明する図である。

【図40】

VCCタグの生成過程を示す図である。

【図41】

生成されたリンクを示す図である。

【図42】

生成されたパスを示す図である。

【図43】

選択されたパスを示す図である。

【図44】

VCCタグの子がBCCタグである場合の例を示す図である。

【図45】

BCCタグが一時的なVCCタグの子として登録された図である。

【図46】

VCCタグの子がSS/SRタグである場合の例を示す図である。

【図47】

SS、SRタグの処理を説明する図である。

【図48】

特異領域SR0の接続パターンの例を示す図である。

【図49】

広義の仮想境界点列を子とする VC タグの生成を示す図である。

【図 5 0】

(a) ～ (d) は、本発明による処理結果を示す図である。

【図 5 1】

本発明をソフトウェアによって実行する場合の構成例を示す図である。

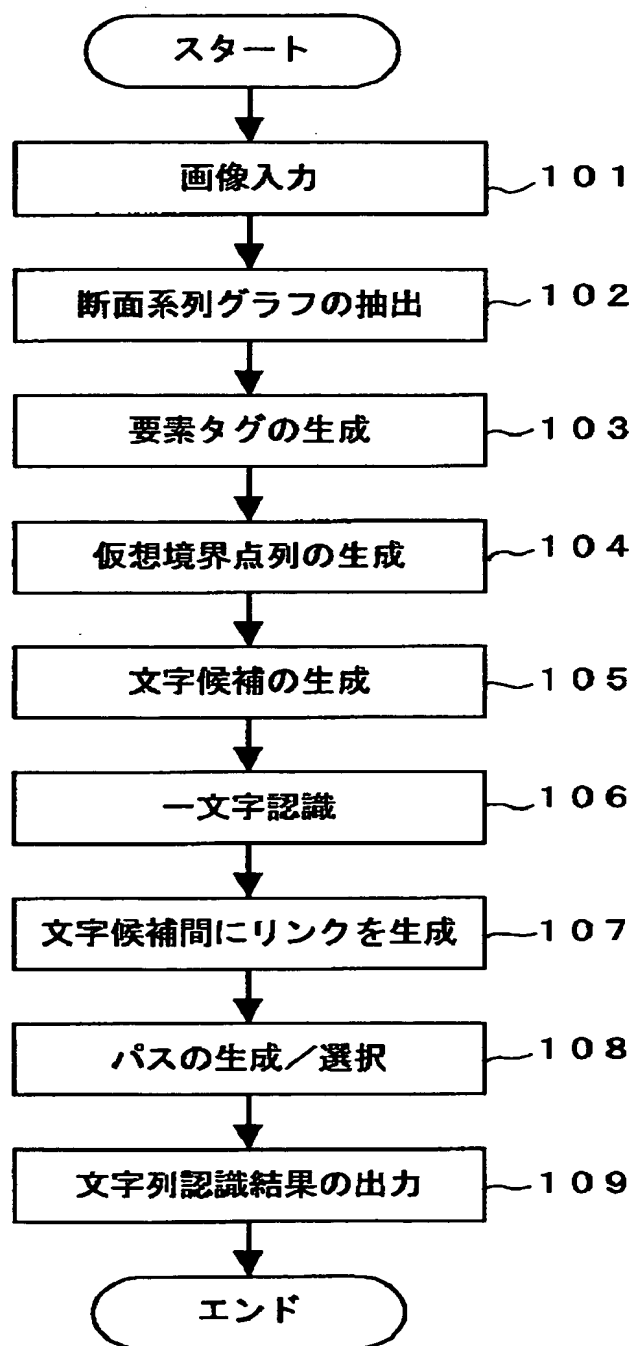
【符号の説明】

- 1 画像入力部
- 2 断面系列処理部
- 3 タグ処理部
- 4 仮想境界点処理部
- 5 メモリ
- 6 リンク生成／パス生成部
- 7 文字認識部
- 8 制御部
- 10 境界点構造体
- 20 境界点列構造体
- 30 断面構造体
- 40 断面系列構造体
- 50 特異領域構造体
- 60 仮想境界点列構造体
- 70 タグ構造体

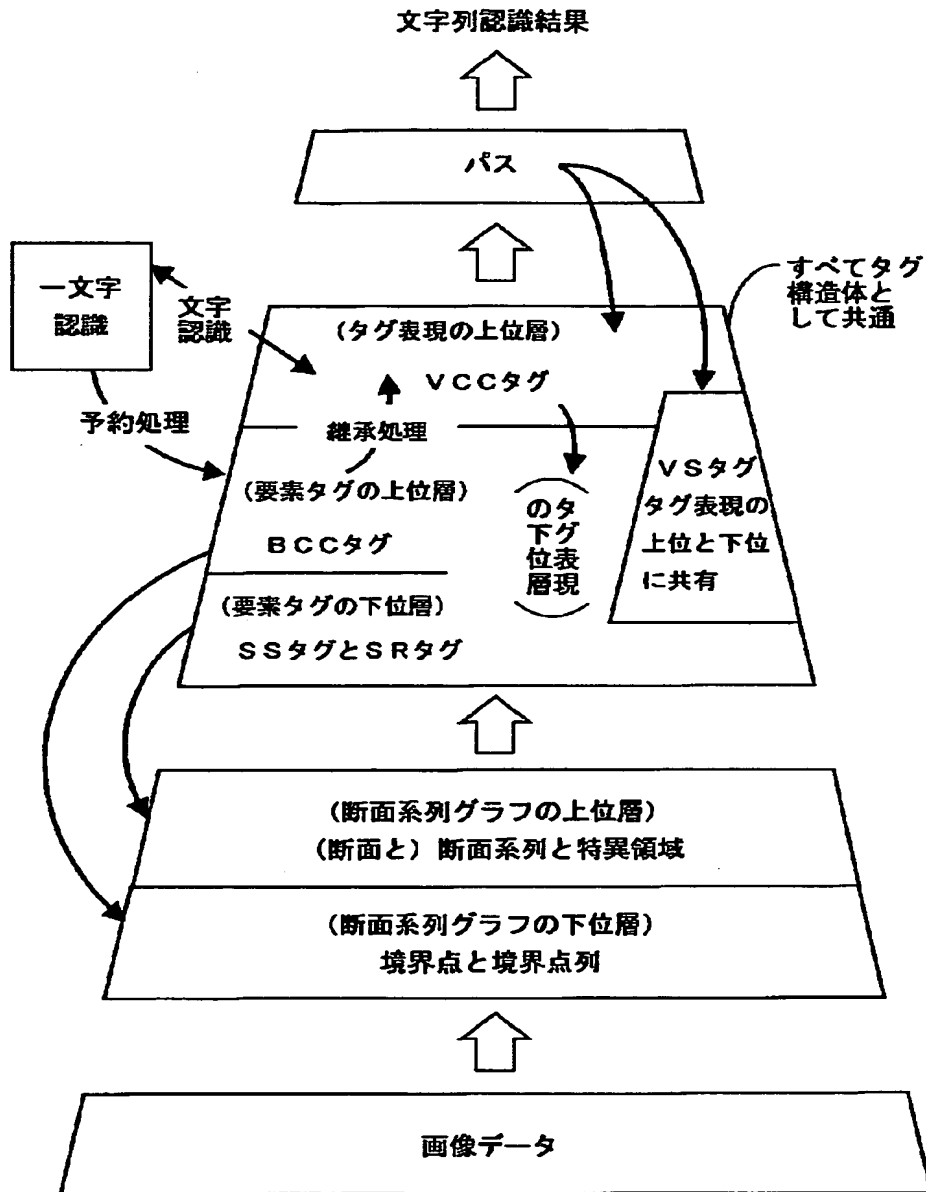
【書類名】

図面

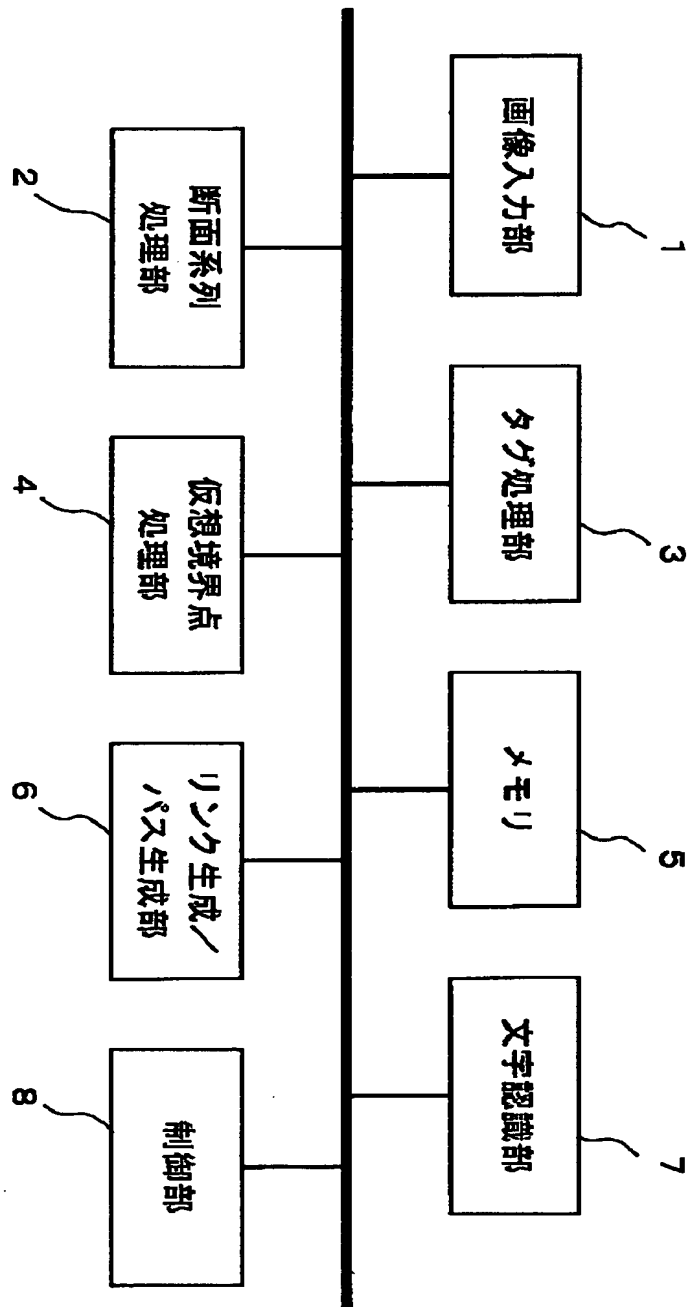
【図 1】



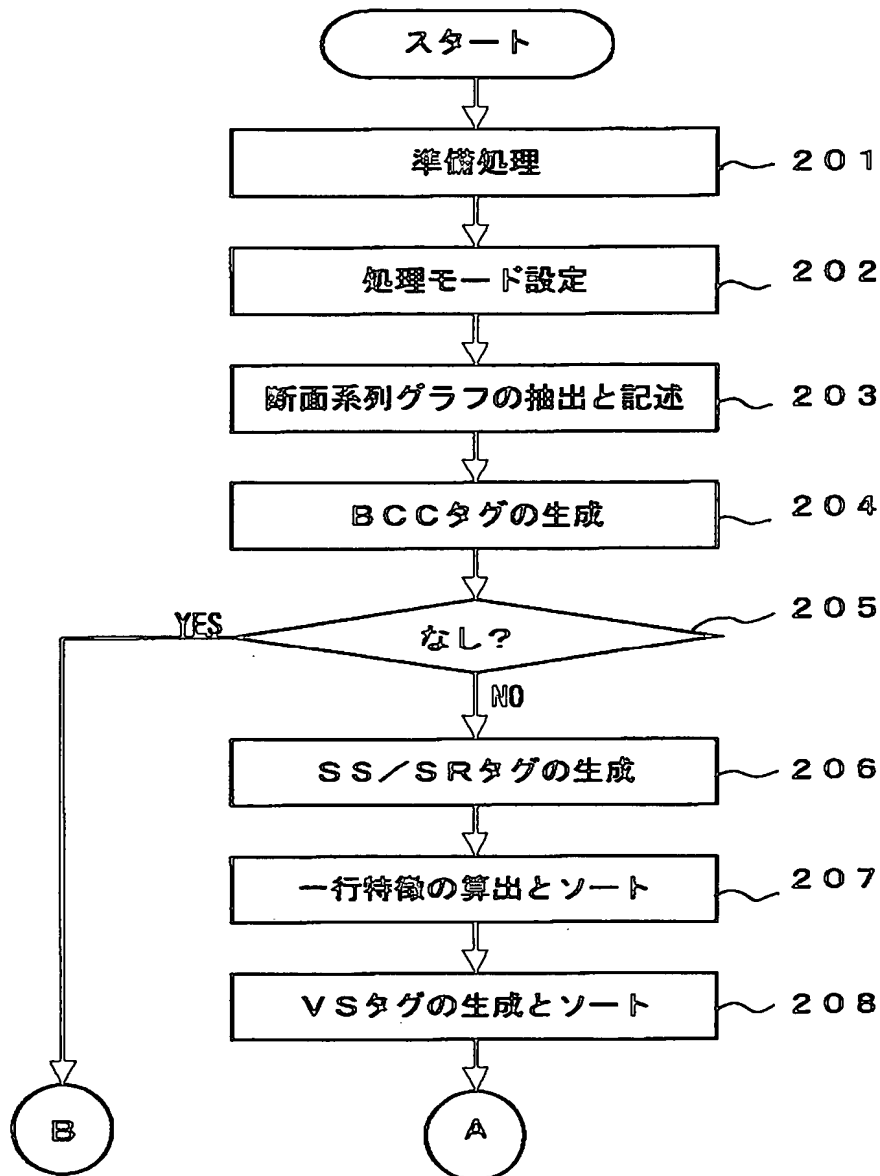
【図 2】



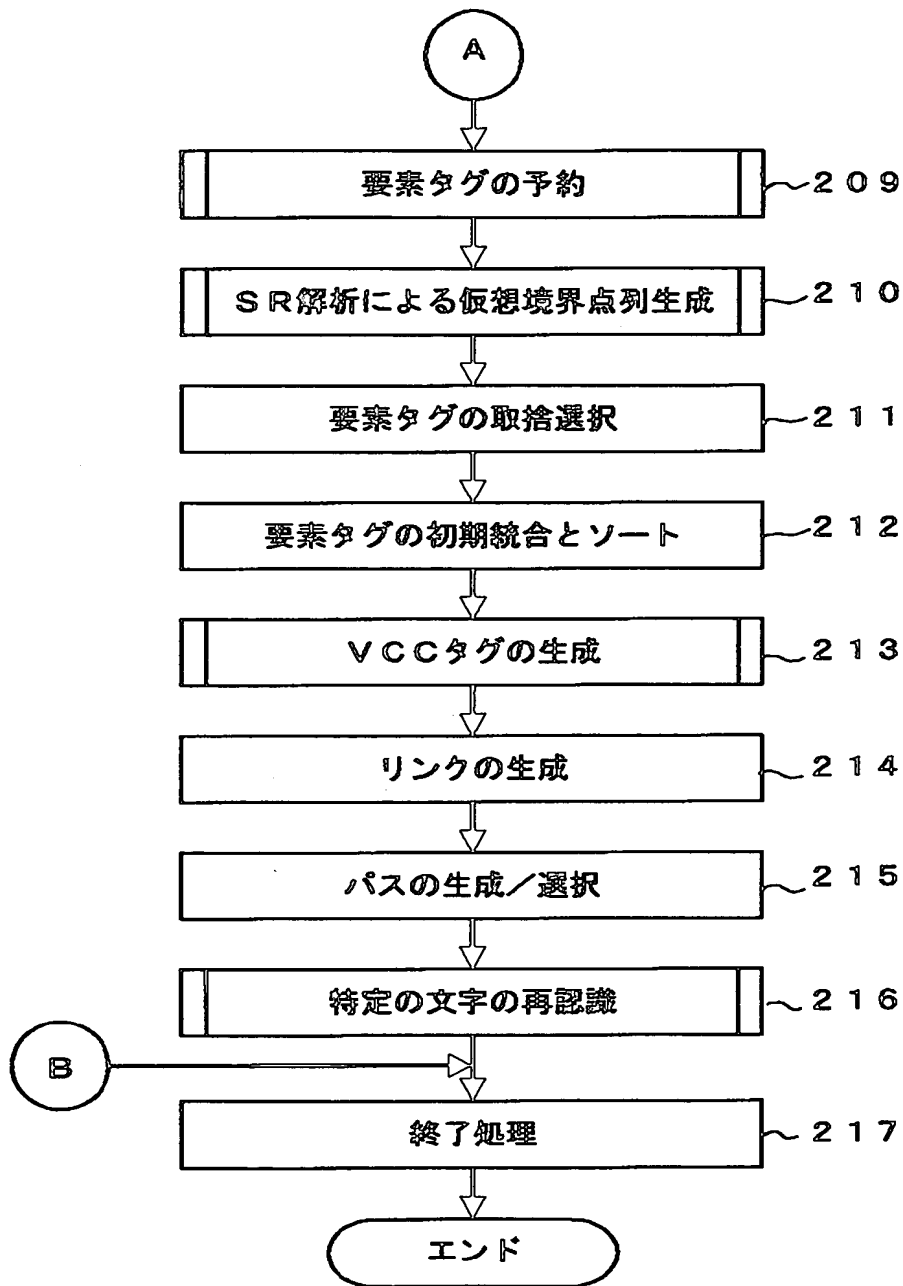
【図 3】



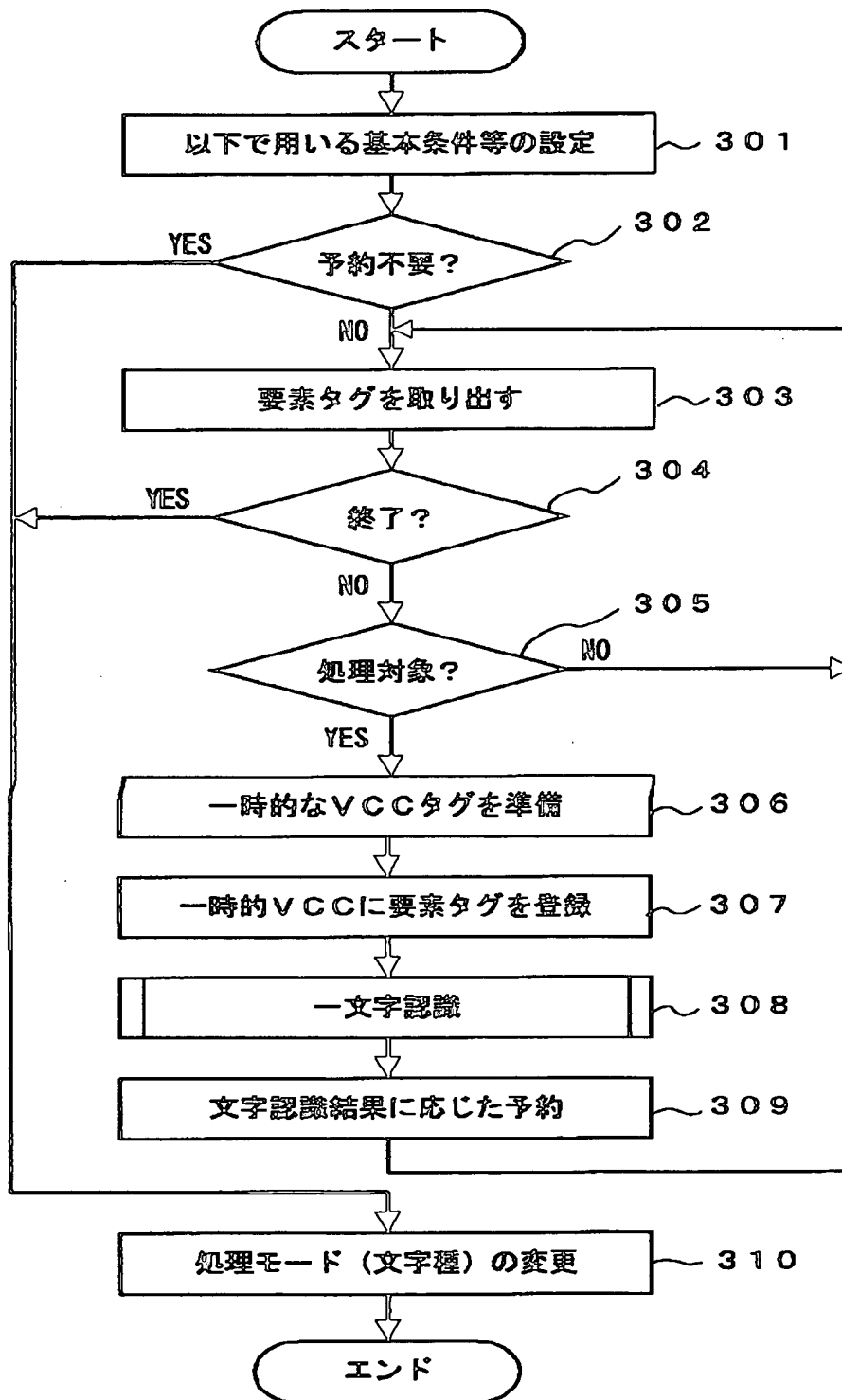
【図 4】



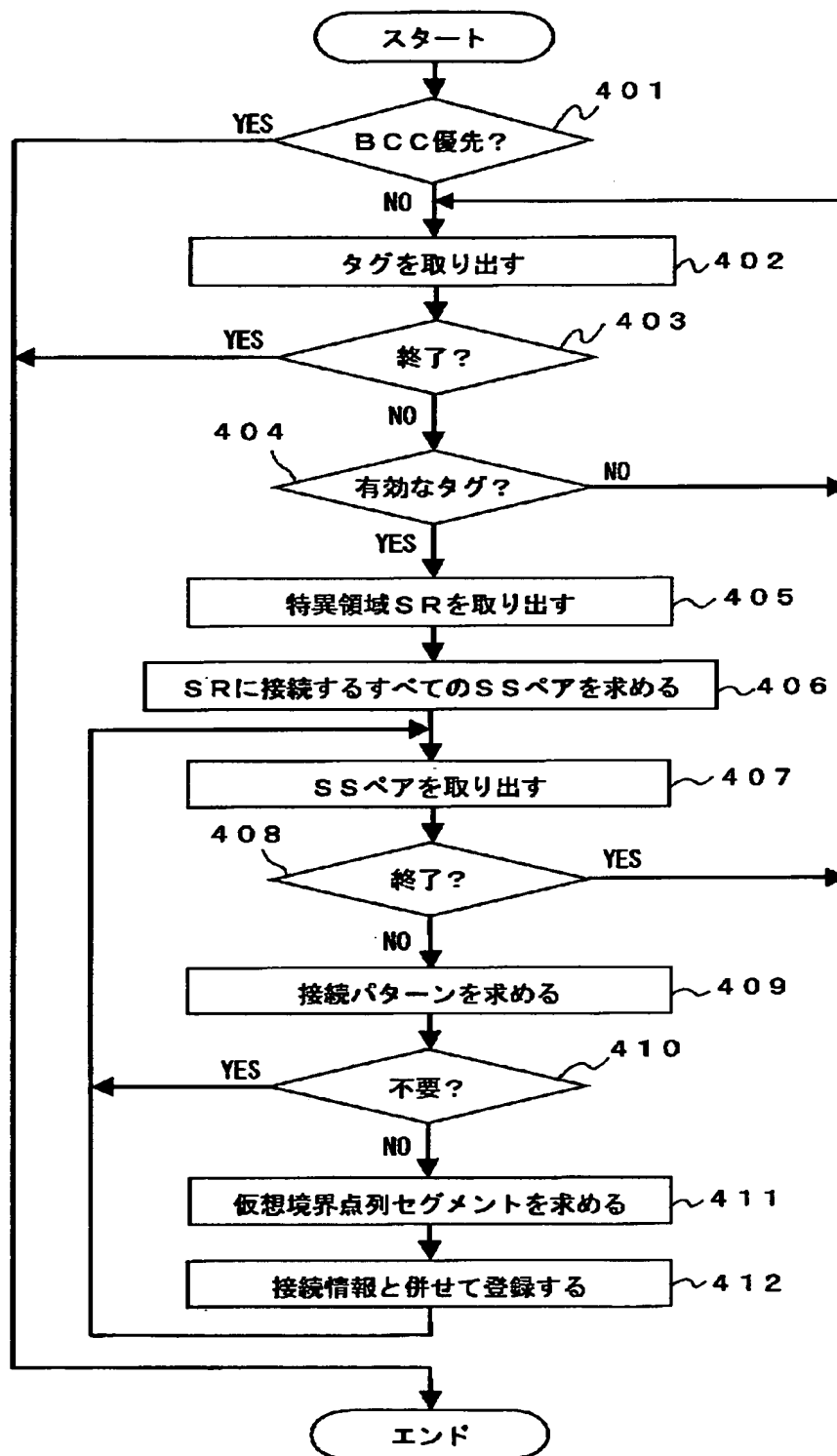
【図 5】



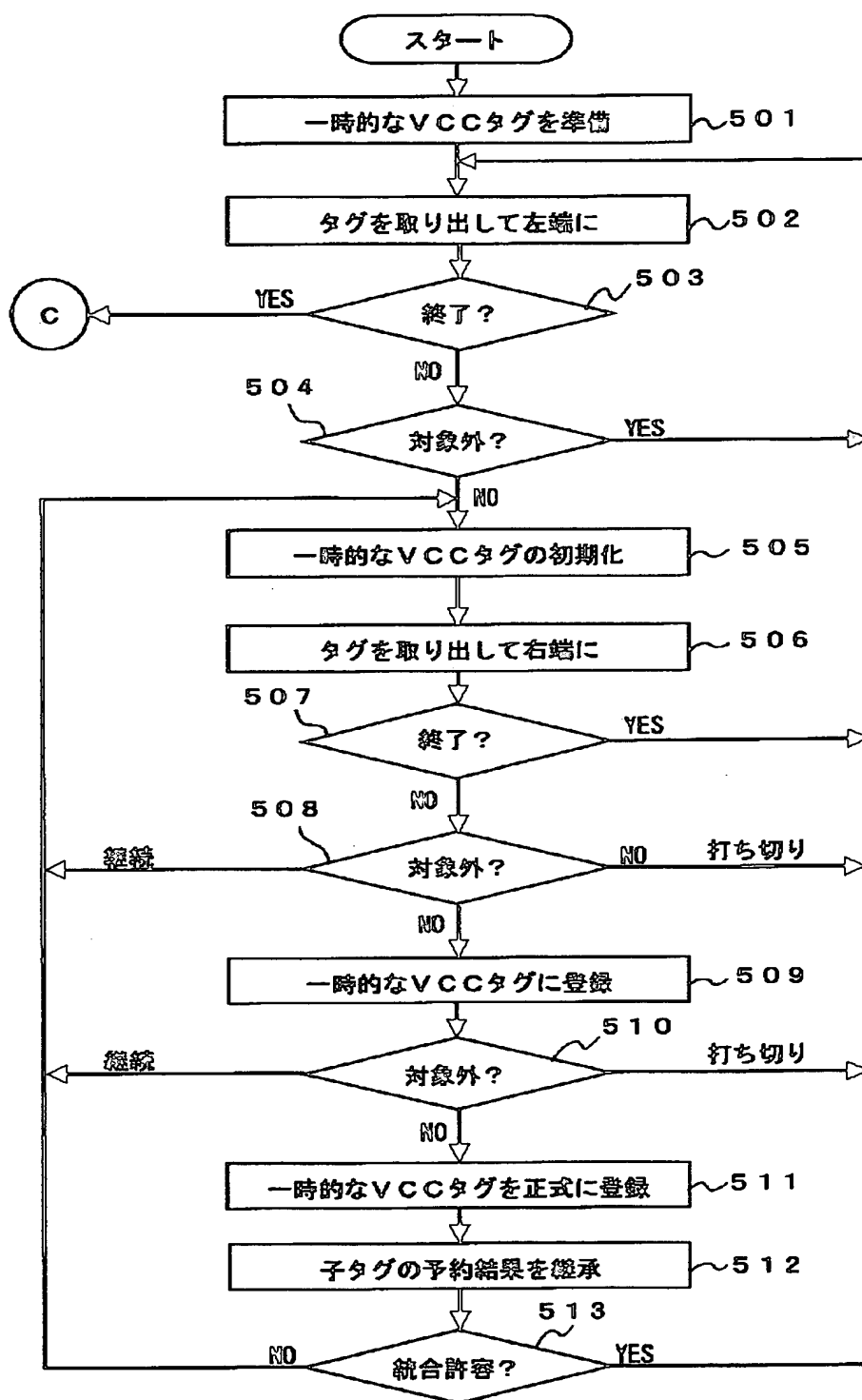
【図 6】



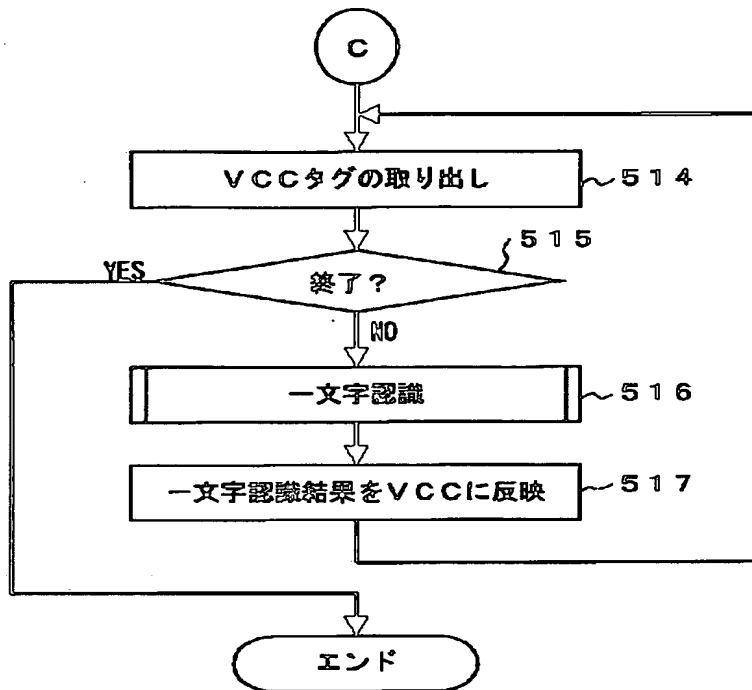
【図 7】



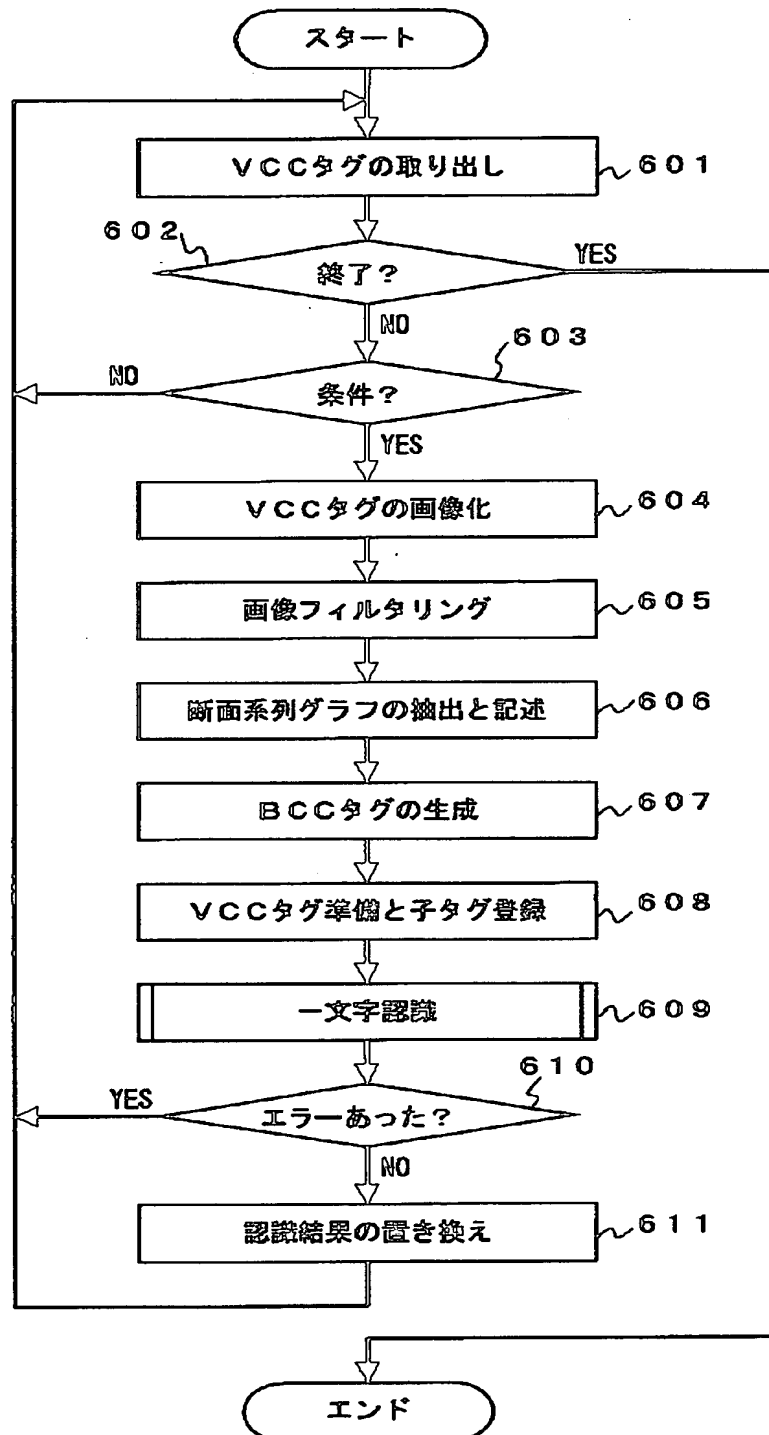
【図 8】



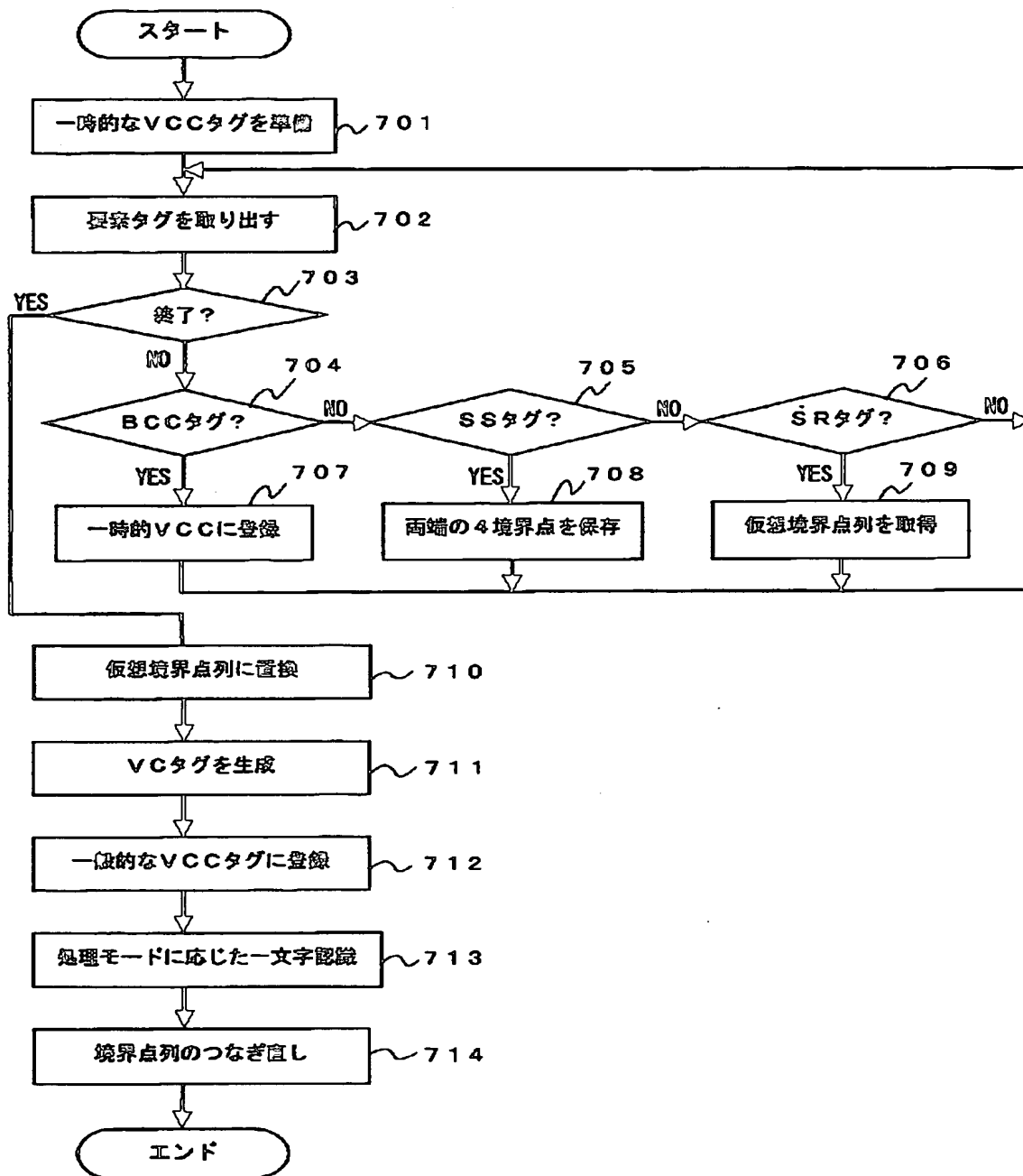
【図9】



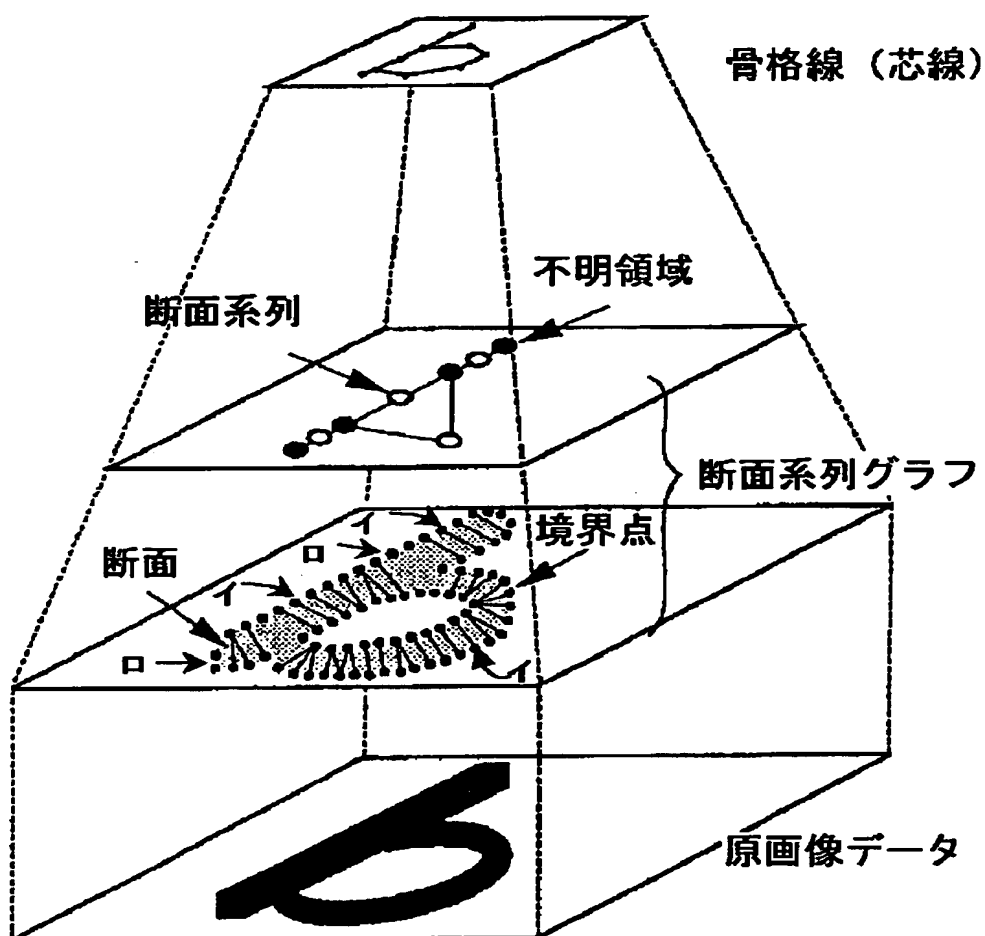
【図10】



【図 11】



【図12】

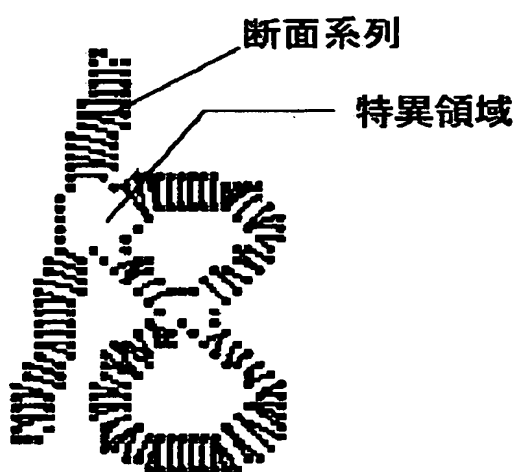


【図13】



文字列画像

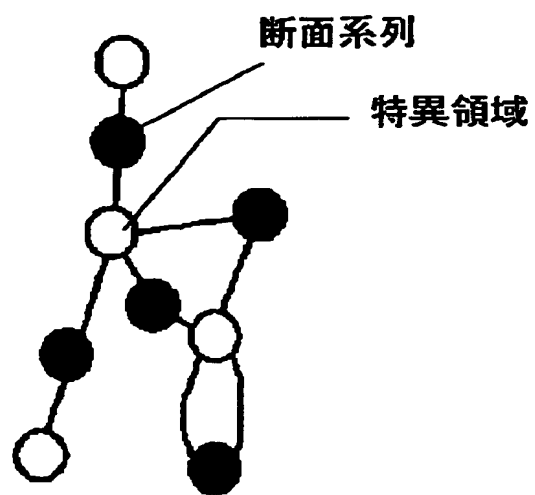
【図14】



境界点と断面

断面系列グラフ

【図 1 5】

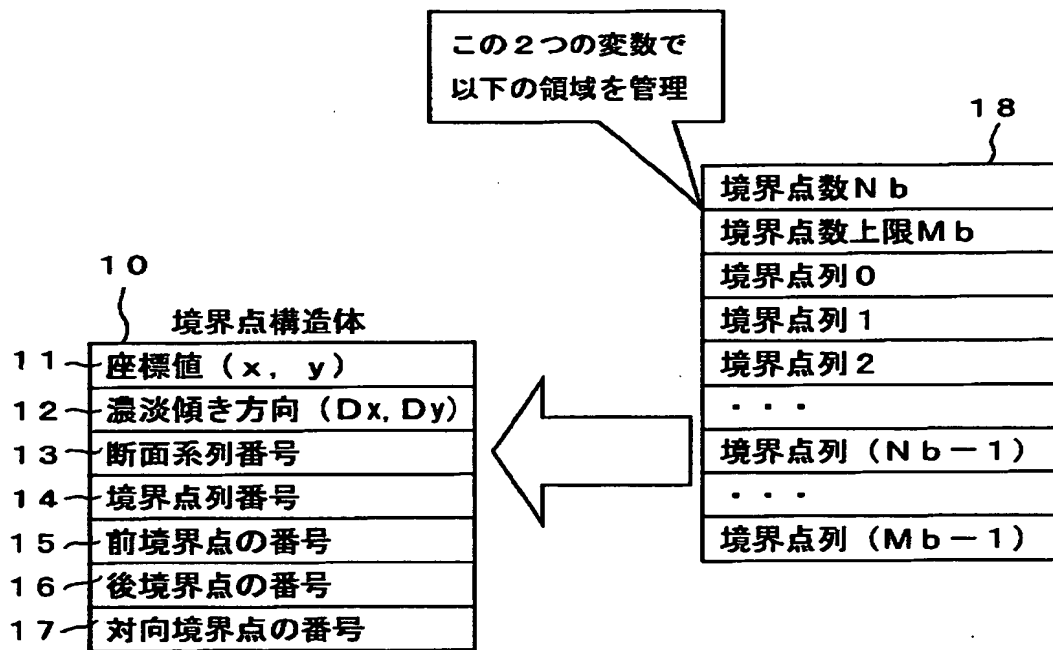


断面系列と特異領域

断面系列グラフ

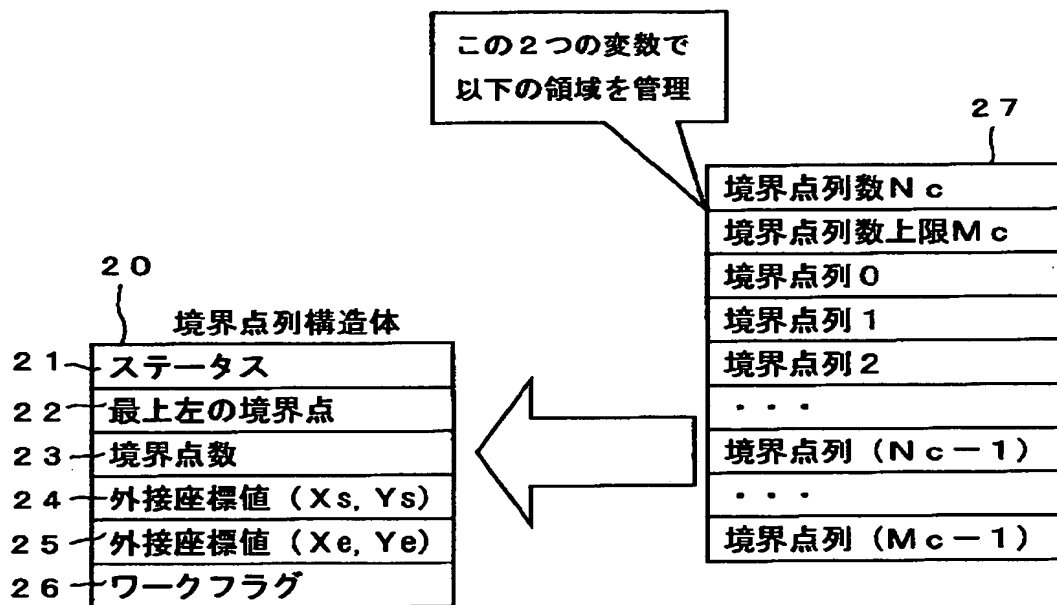
【図 1 6】

境界点の表現形式



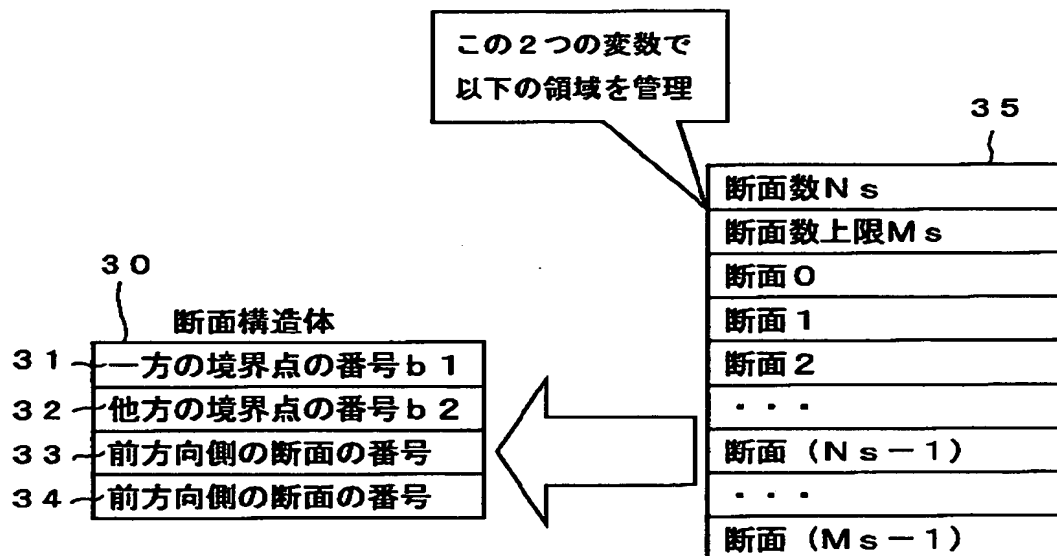
【図 1 7】

境界点列の表現形式



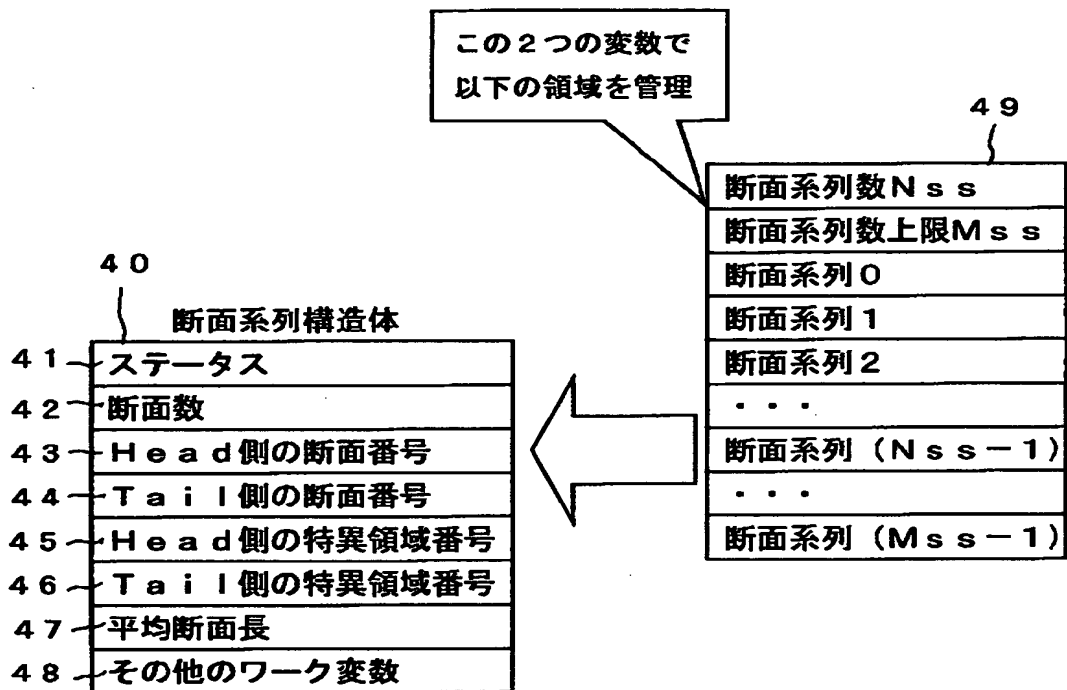
【図 1 8】

断面の表現形式



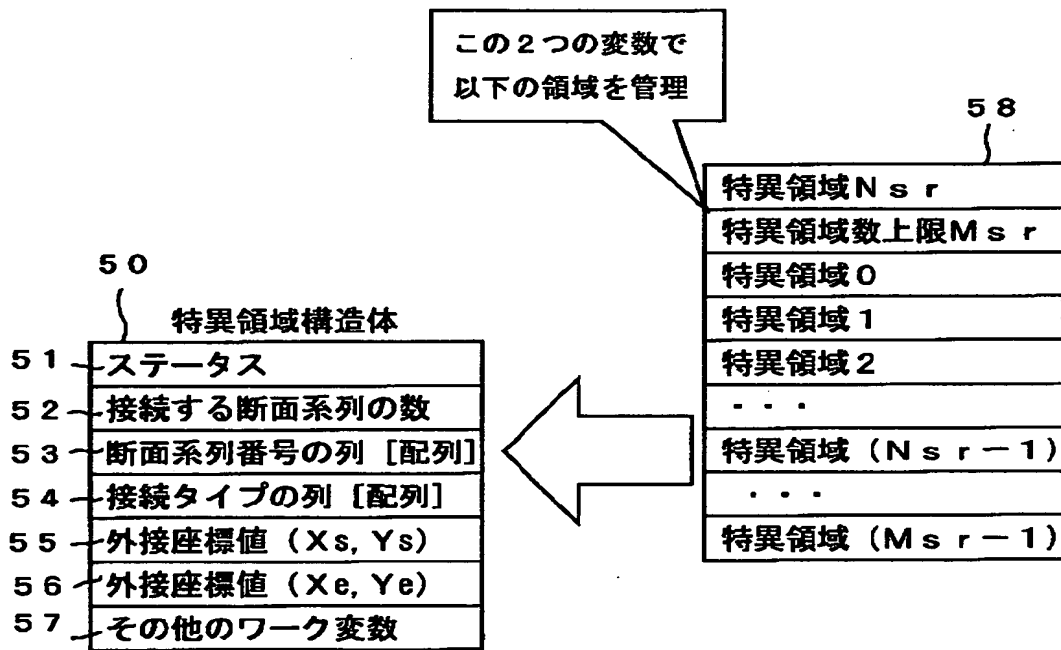
【図 1 9】

断面系列の表現形式



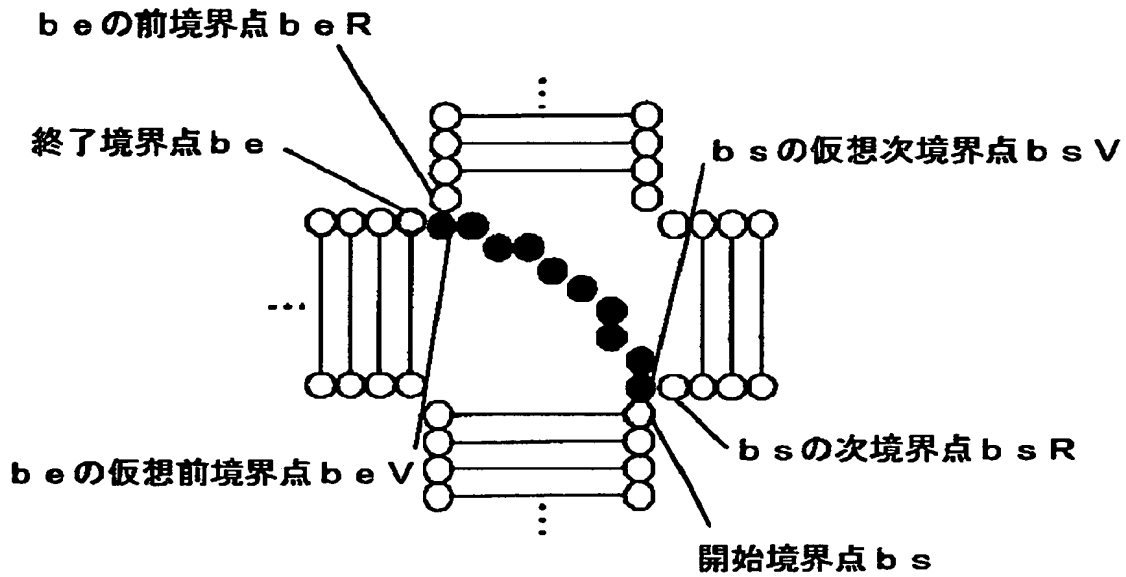
【図 2 0】

特異領域の表現形式



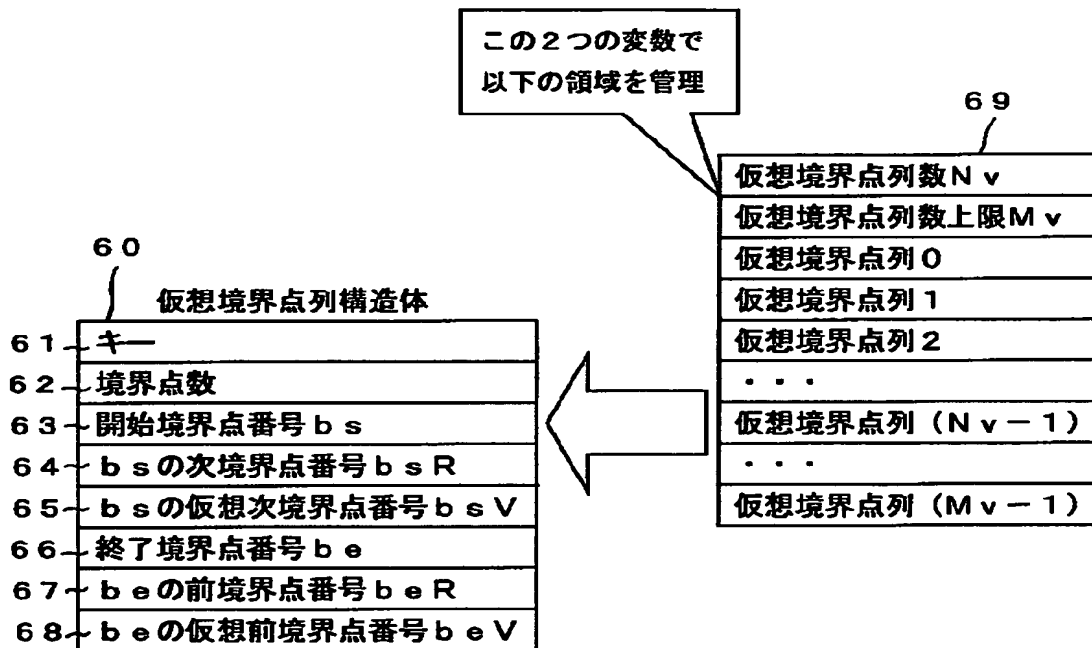
【図 2 1】

仮想境界点列セグメントの説明



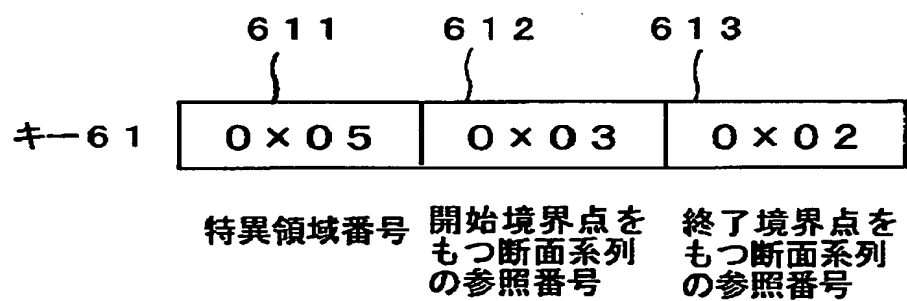
【図 2 2】

仮想境界点列の表現形式



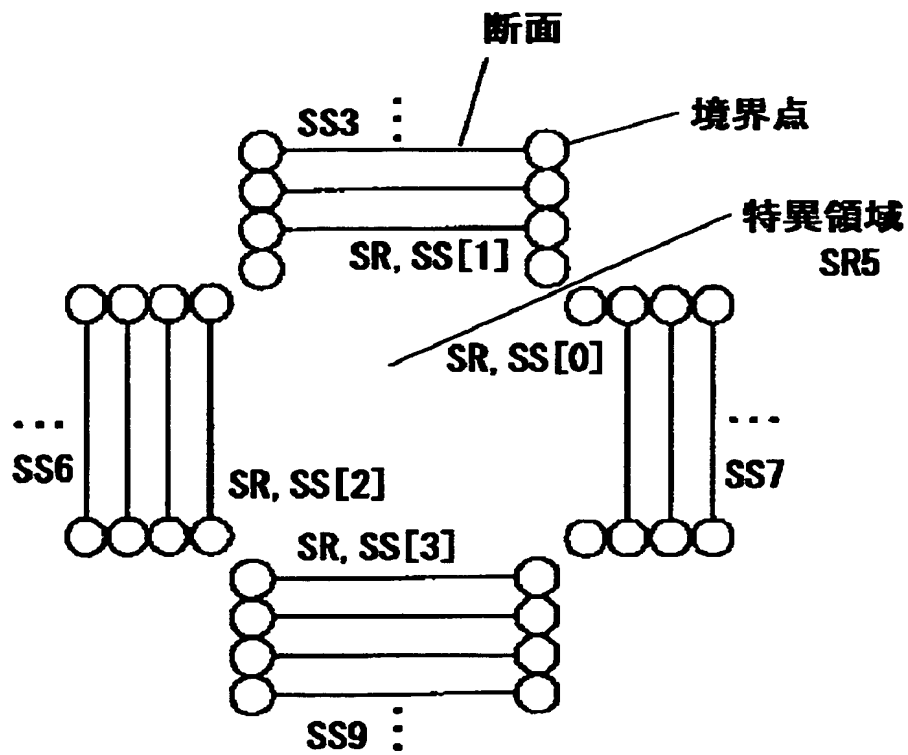
【図 2 3】

特異領域の接続パターンの説明



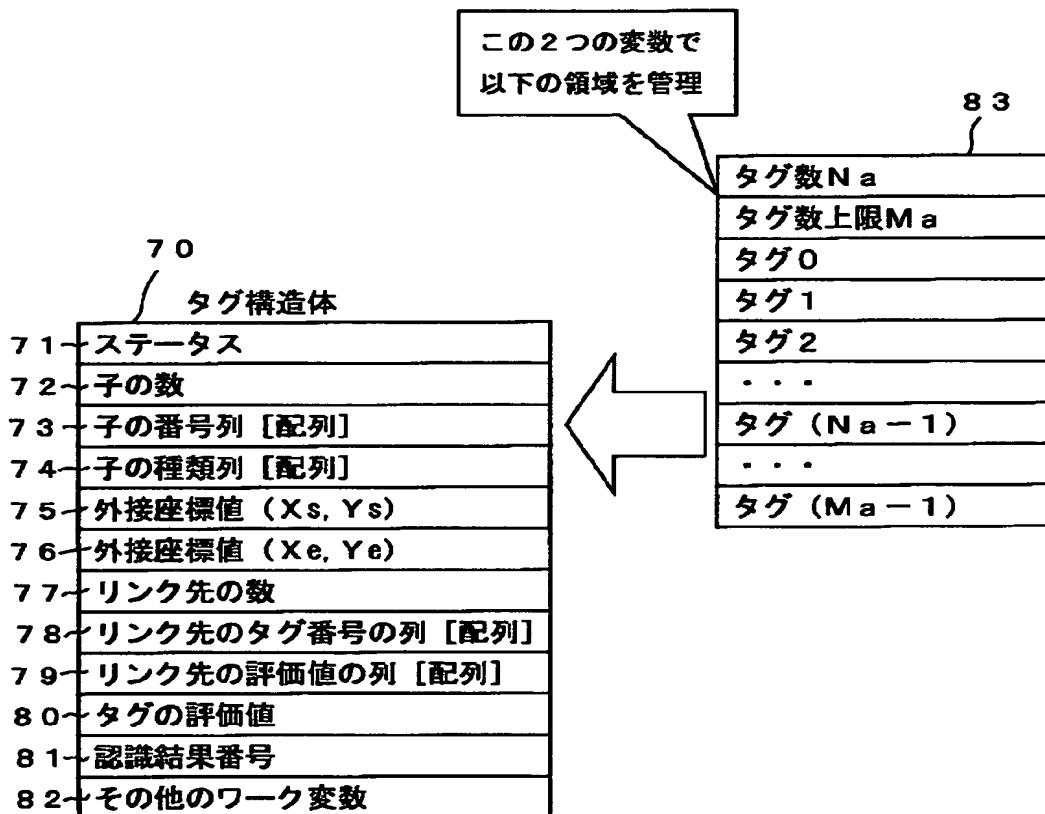
【図 2 4】

特異領域の接続の様子



【図 2 5】

タグの表現形式



【図 2 6】

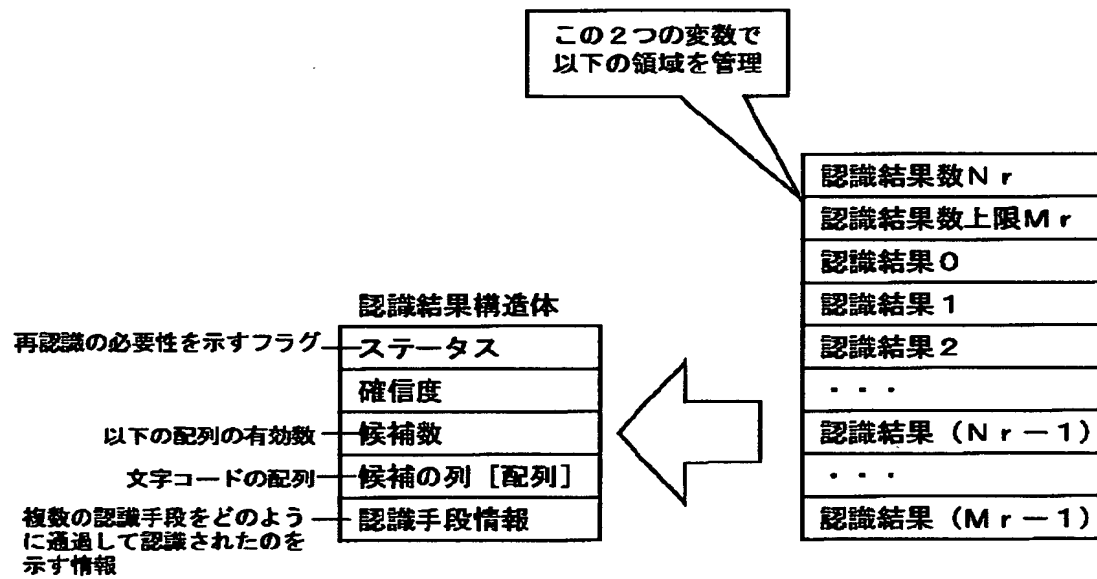
タグを制御する構造体

タグ制御構造体		
文字種、層優先度など	処理モード変数群	
	境界点配列へのポインタ	図 1 6 へ
	境界点列配列へのポインタ	図 1 7 へ
	断面配列へのポインタ	図 1 8 へ
	断面系列配列へのポインタ	図 1 9 へ
	特異領域配列へのポインタ	図 2 0 へ
	仮想境界点列配列へのポインタ	図 2 2 へ
	タグ配列へのポインタ	図 2 5 へ
図 2 9 参照のこと	タグのソート結果配列 [配列]	
	パス (パスタグ番号の配列) [配列]	
文字列の外接矩形座標、 幅、高さ、総境界点数など	認識結果配列へのポインタ	図 2 7 へ
	文字列の幾何的情報等の変数群	
	特徴あるタグの番号群 [配列]	

図 2 8 で示される特定のタグの番号のこと
タグの処理範囲を限定
するのに用いられる

【図 2 7】

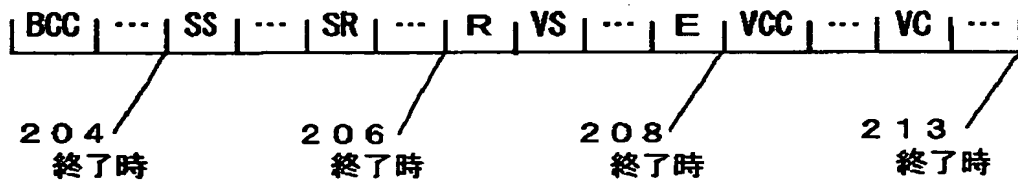
認識結果の表現形式



【図 2 8】

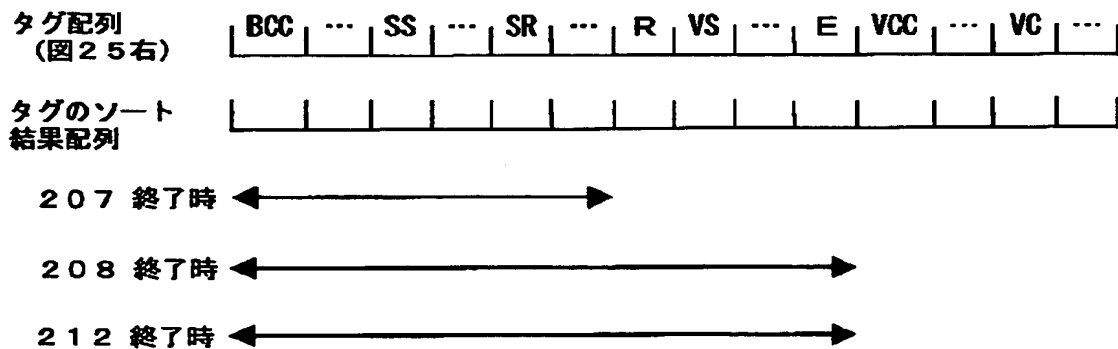
タグ配列が使われる様子

BCC : 通常の BCC タグ
 SS : SS タグ
 SR : SR タグ
 R : ルートタグ (VS タグの一種)
 VS : VS タグ
 E : エンドタグ (VS タグの一種)
 VCC : VCC タグ
 VC : 仮想境界点列で一部置き換えられている BCC タグ



【図 2 9】

タグ配列とソート結果配列の様子



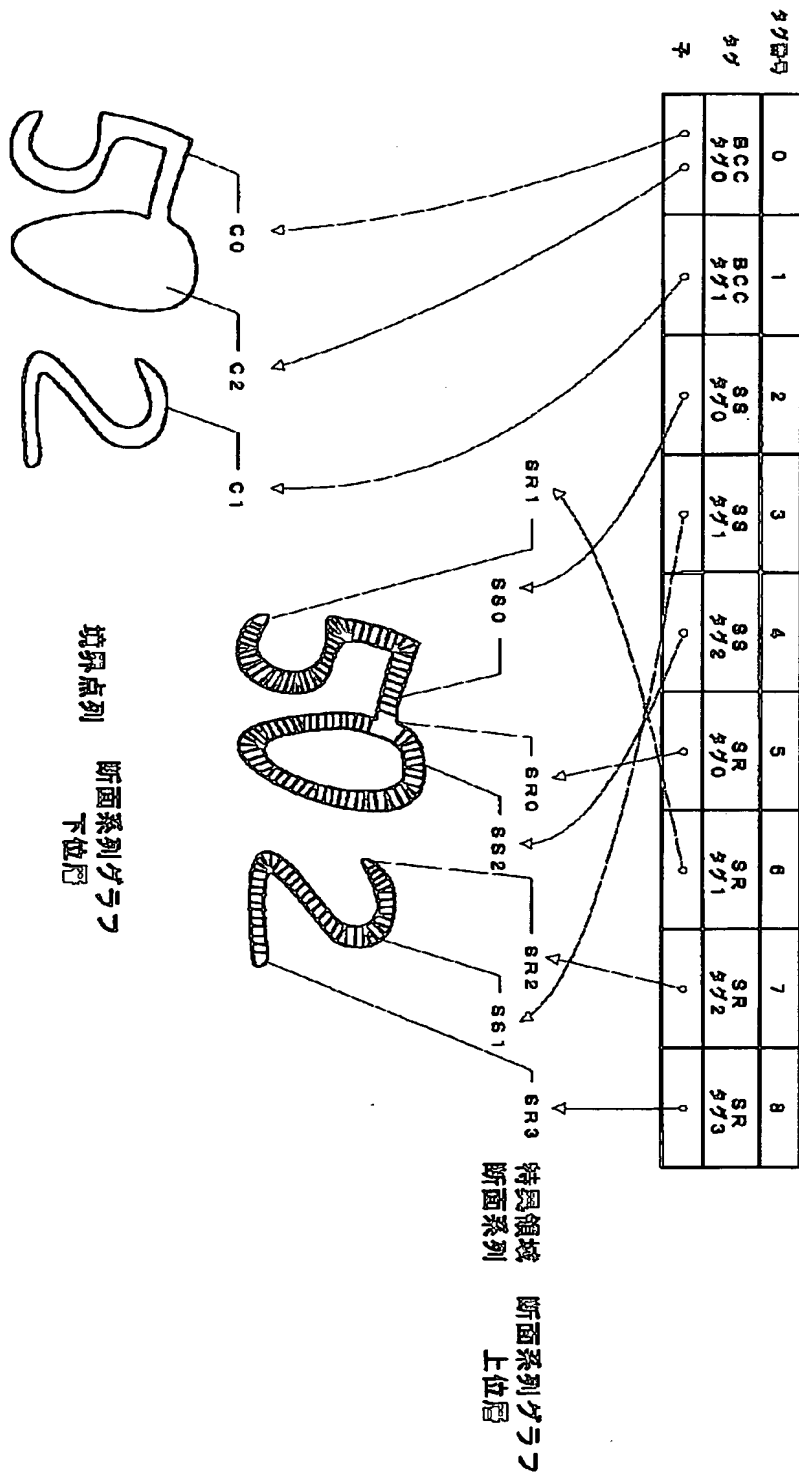


【図 3 0】

502

画像データ

【図 31】



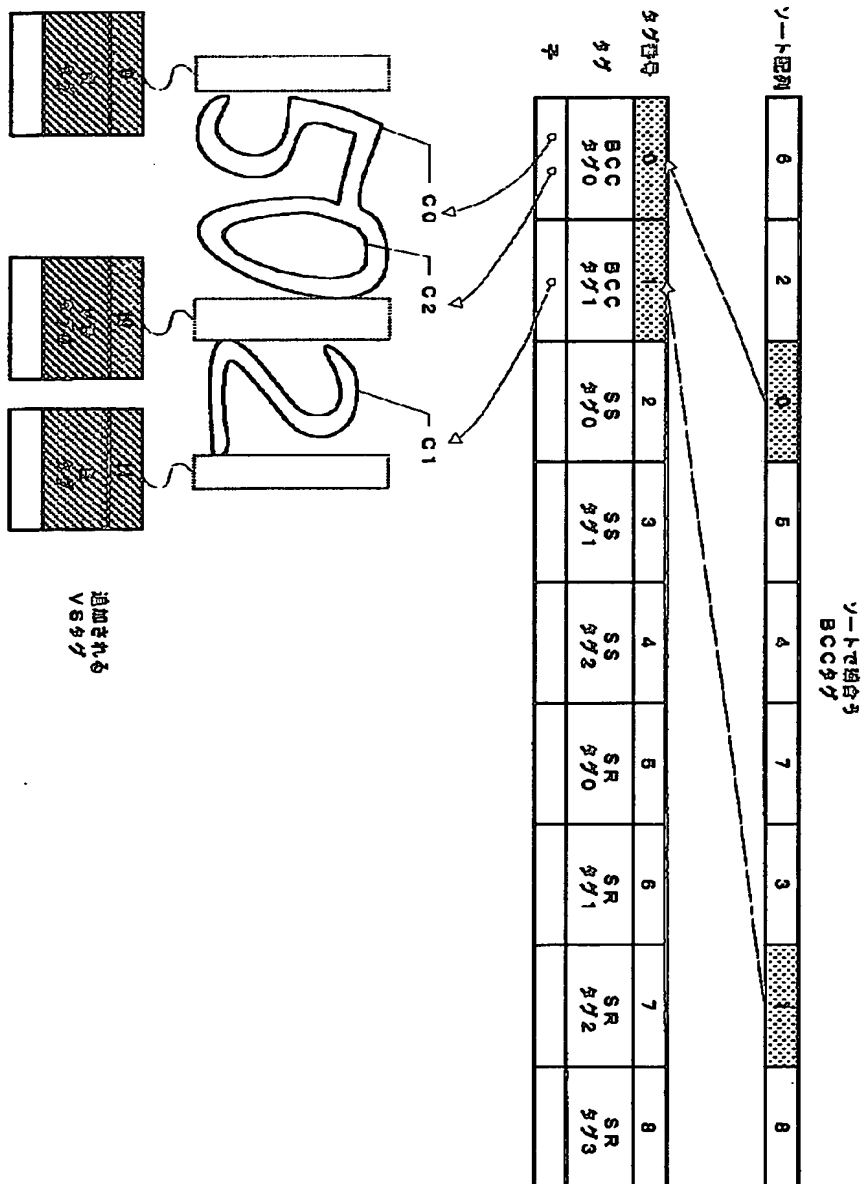
【図 3 2】

タグ番号

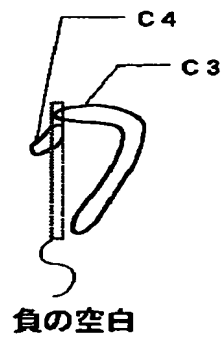
シート配列

8	2	0	5	4	7	3	1	8
---	---	---	---	---	---	---	---	---

【図 33】



【図 3 4】



負の空白の説明

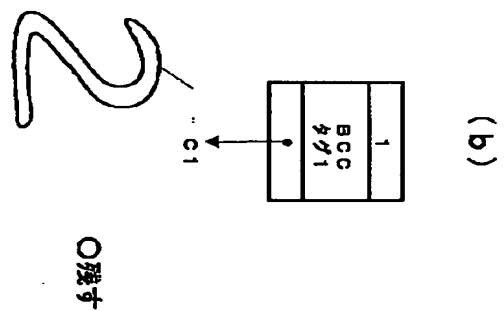
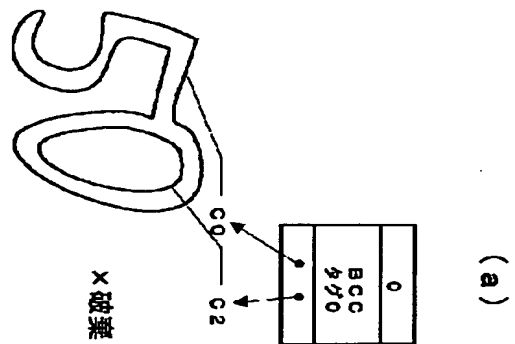
【図 35】

VSタグを含めたシート結果

シート区別	5	6	2	0	5	4	5	7	3	1	8	1
タグ番号	0	1	2	3	4	5	6	7	8	9	10	11
タグ	BCC タグ0	BCC タグ1	SS タグ0	SS タグ1	SS タグ2	SR タグ0	SR タグ1	SR タグ2	SR タグ3	SS タグ4	SS タグ5	SS タグ6
子												

追加された
VSタグ

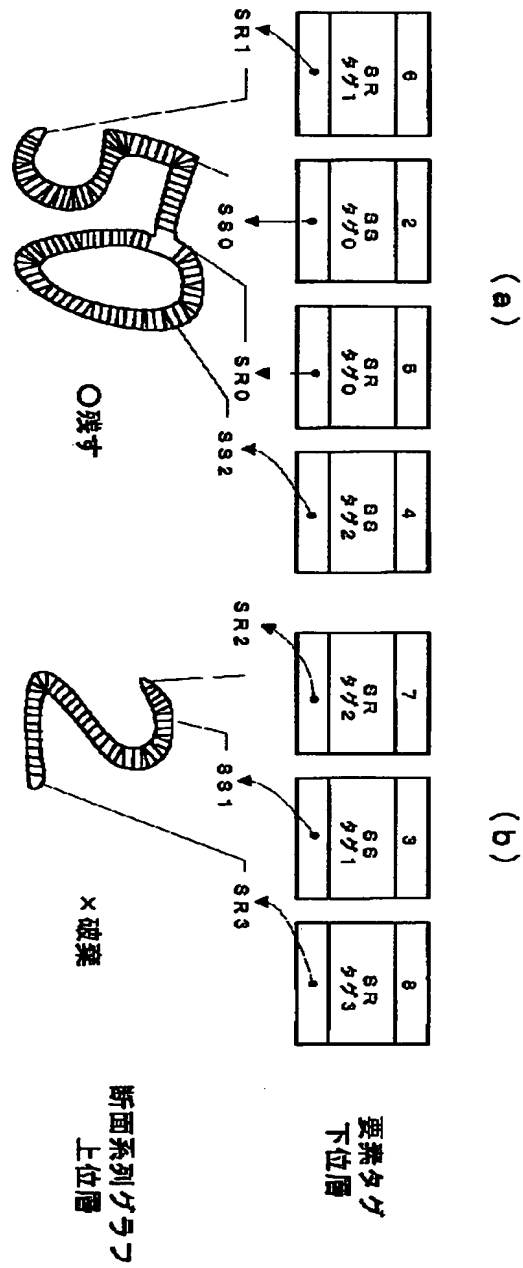
【図 3 6】



要素タタ
上位層

断面系列グラフ
下位層

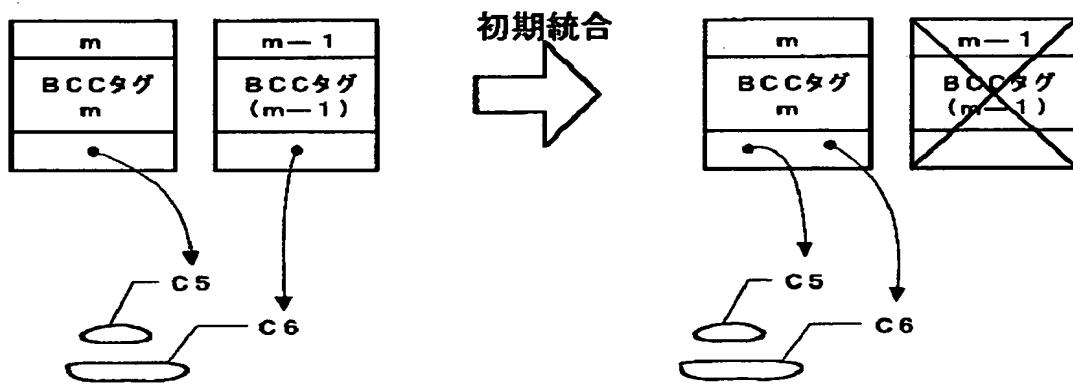
【図37】



【図 3 8】

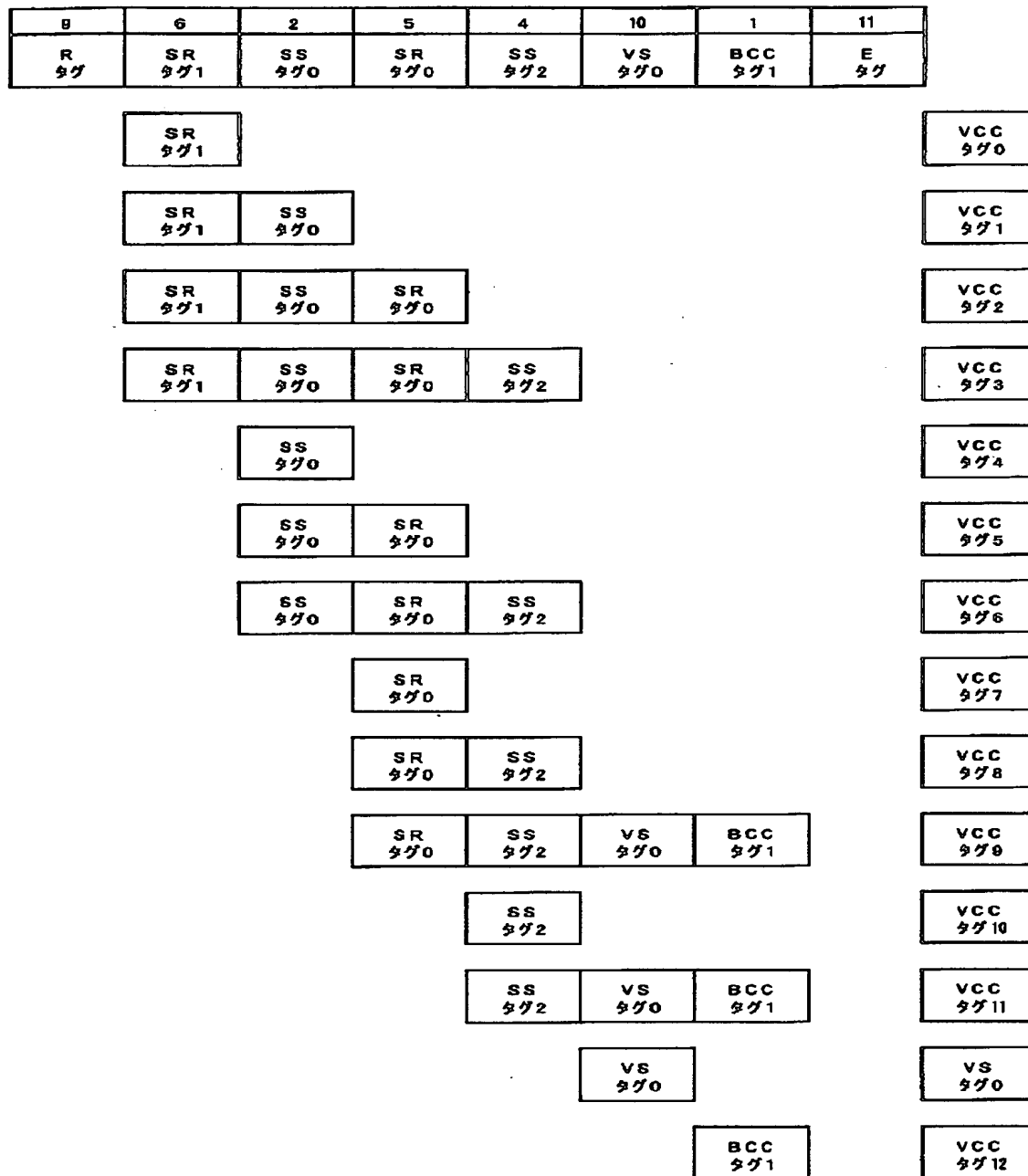
シート配列		9	6	2	0	5	4	10	7	3	1	8	11
タイ番号	0												
	BCC 570												
タイ	1	BCC 571											
			SS 570										
子	2												
				SS 571									
	3												
					SS 572								
	4												
						SR 570							
	5												
							SR 571						
	6												
								SR 572					
	7												
									SR 573				
	8												
										R 57			
	9												
											VS 570		
	10												
												E 57	
	11												

【図 3 9】



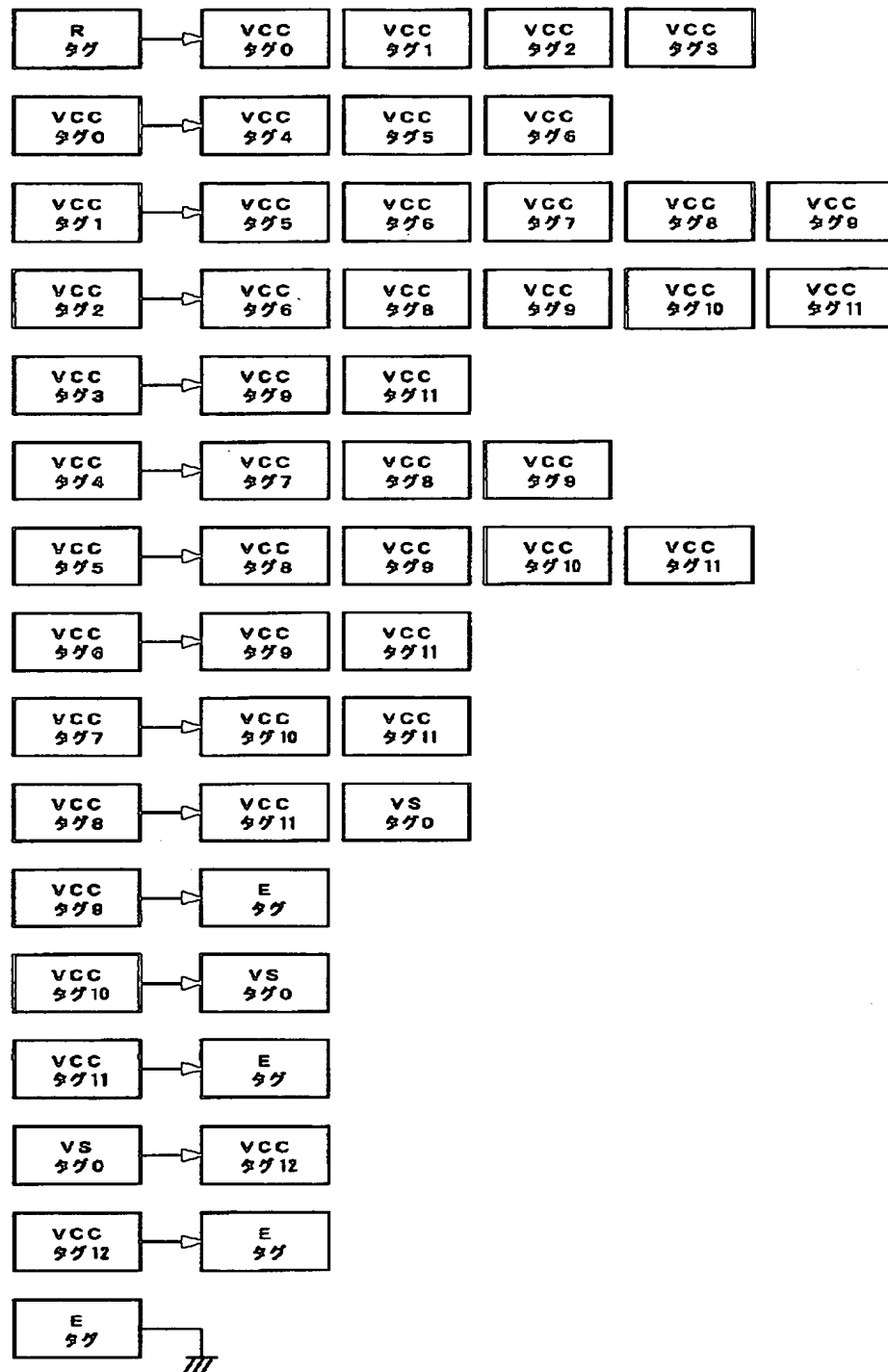
要素タグの
初期統合の説明

【図40】



VCCタグの生成

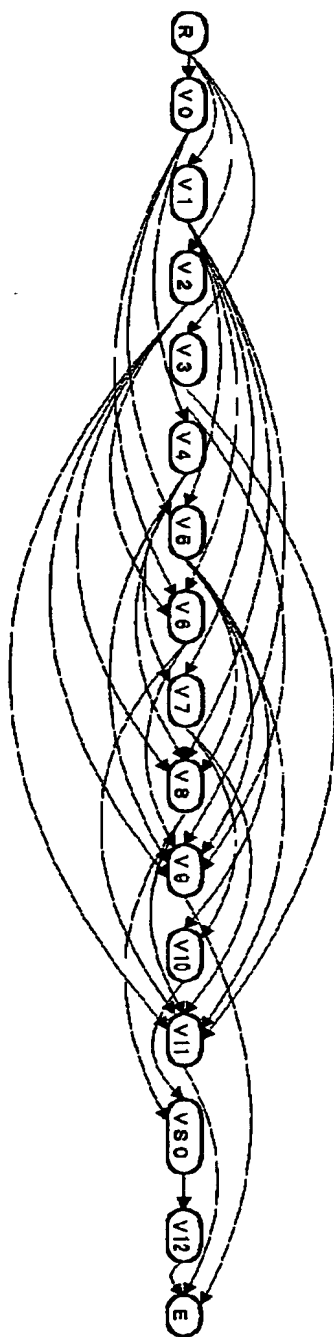
【図41】



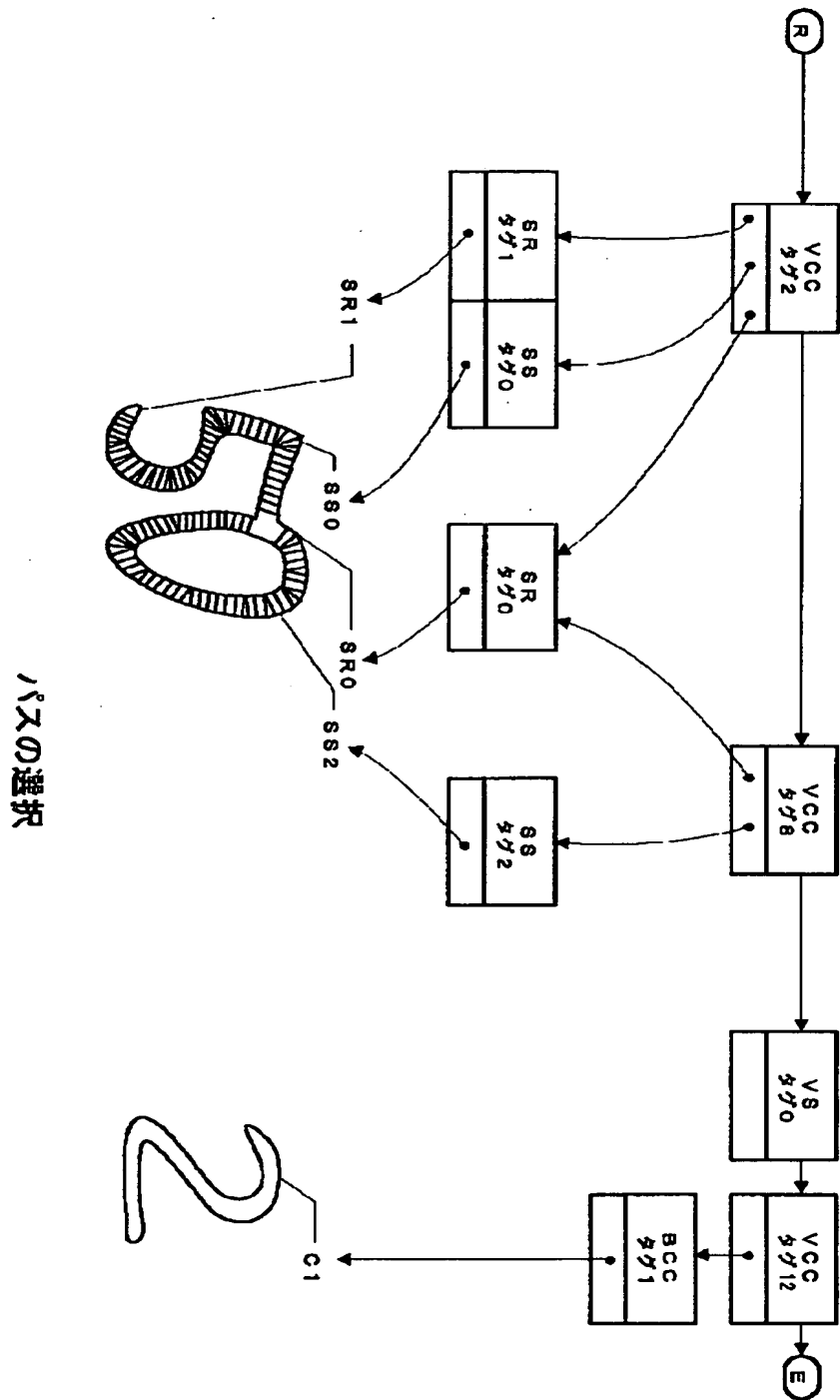
リンクの生成

【図 4 2】

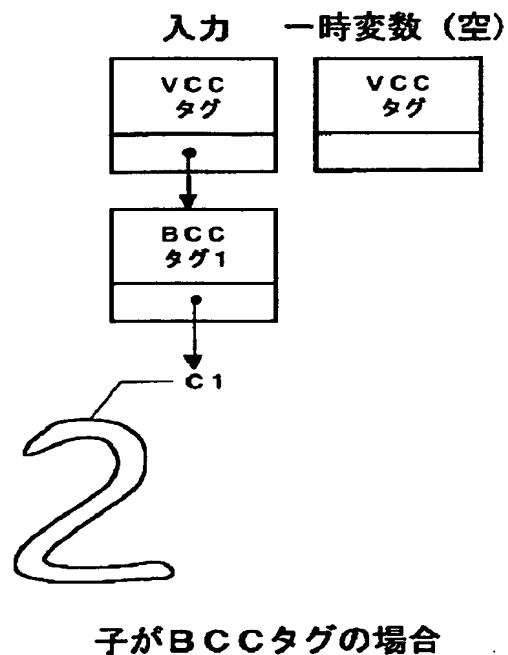
パスの生成



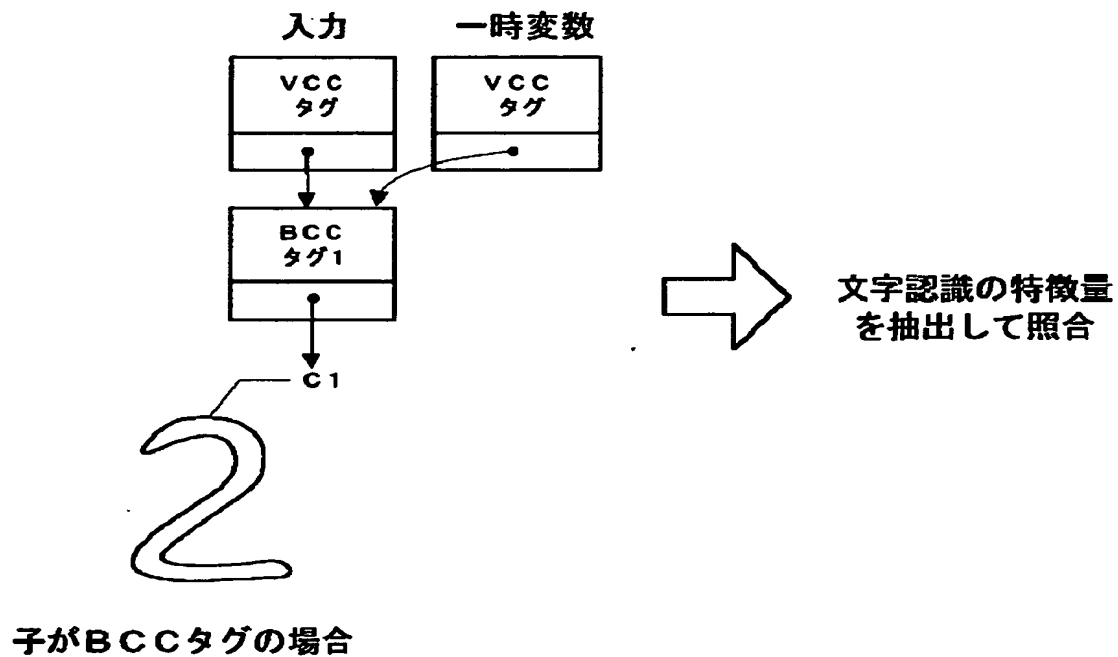
【図 4 3】



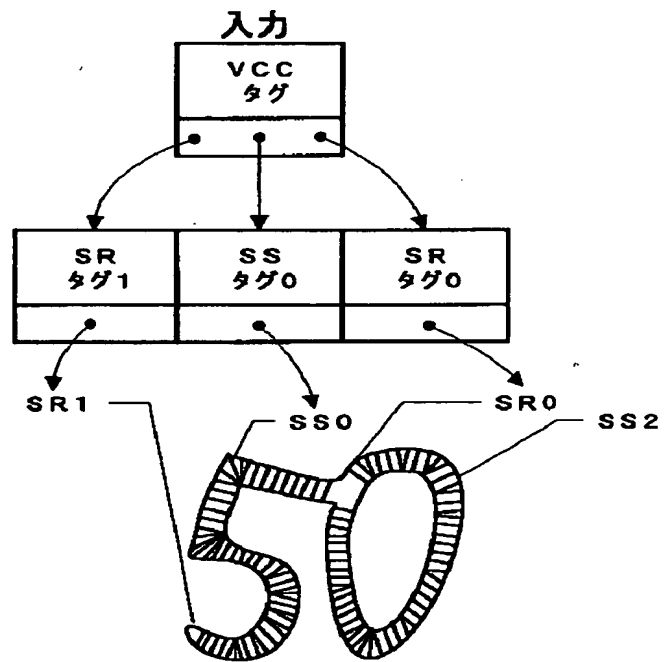
【図 4 4】



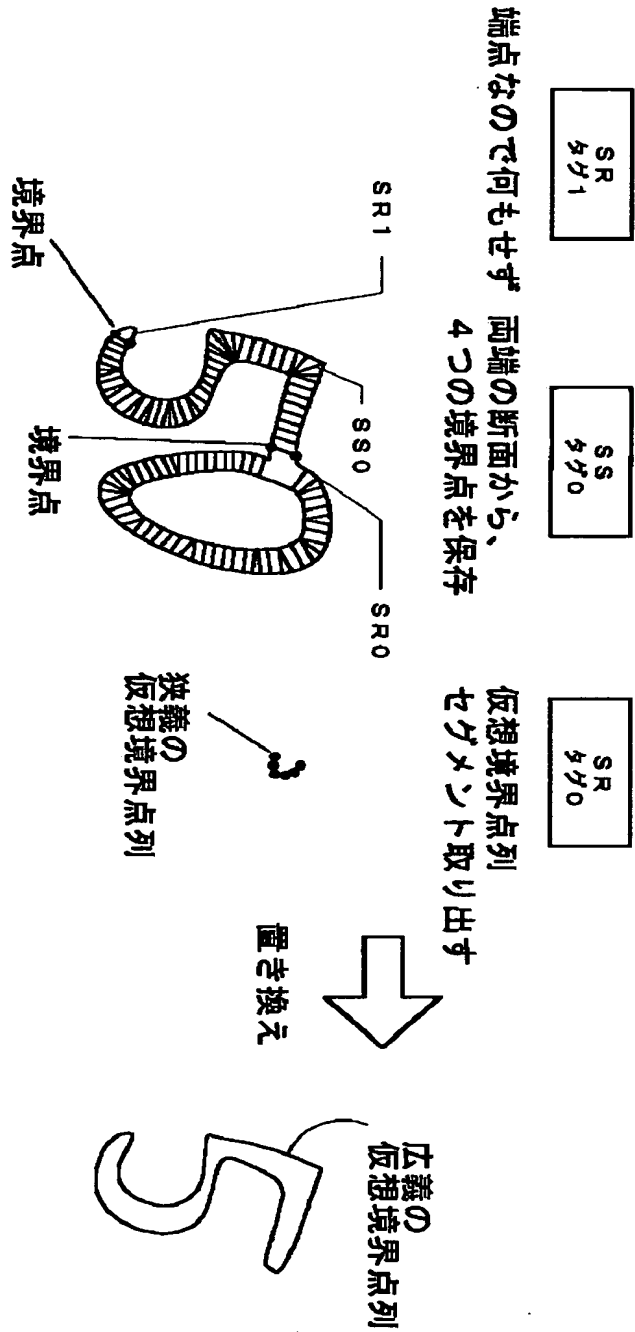
【図 4 5】



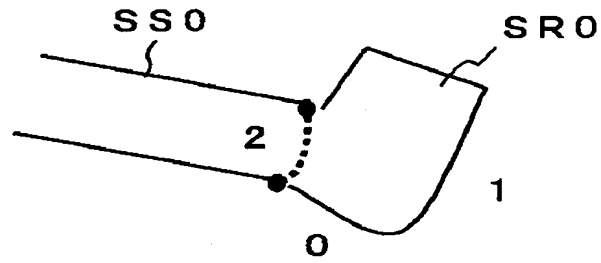
【図 4 6】



【図 47】



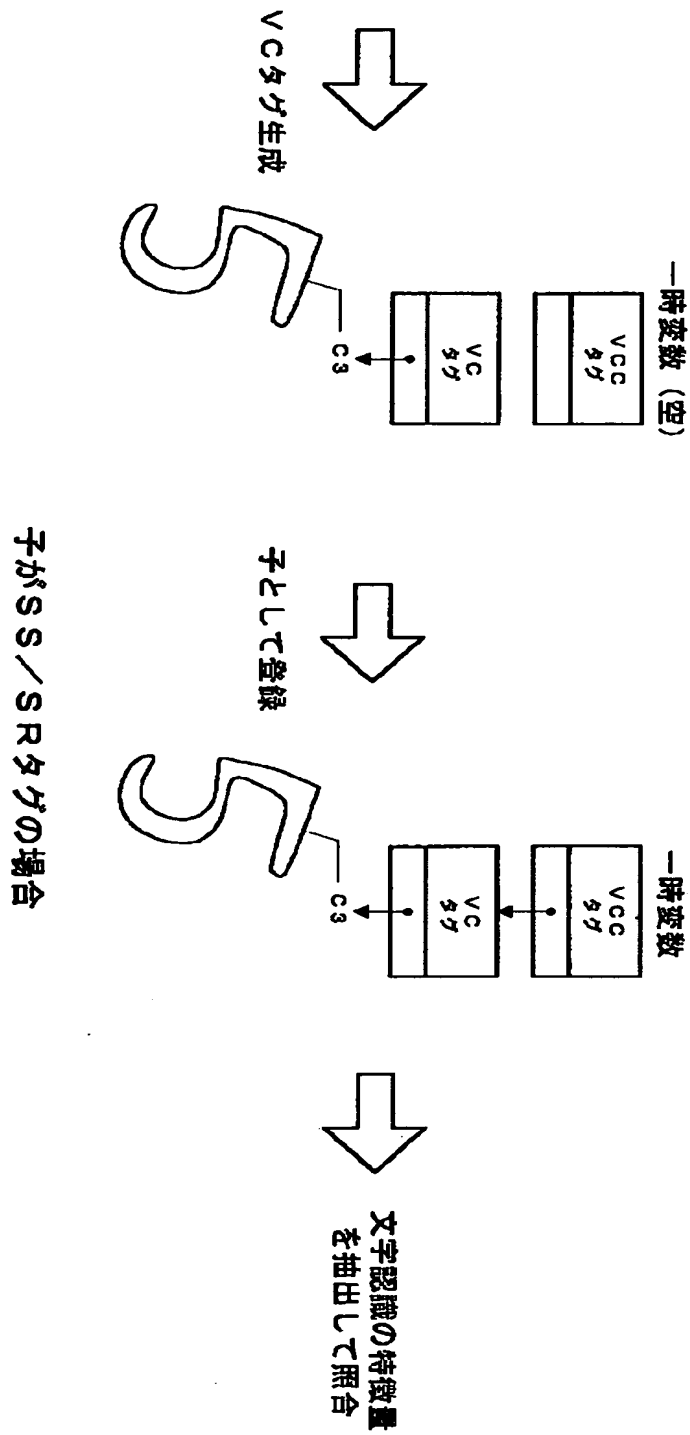
【図 4 8】



キ一

SR0	開始 2	終了 2
-----	------	------

【図 49】



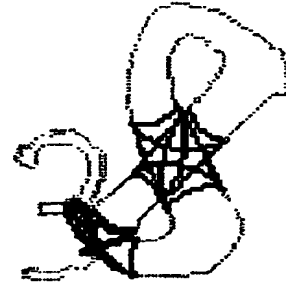
【図50】



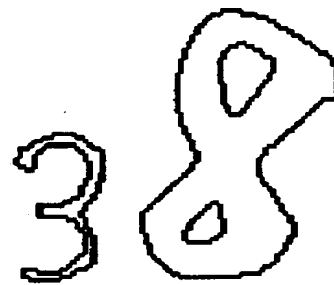
(a) 画像



(b) 断面系列グラフ

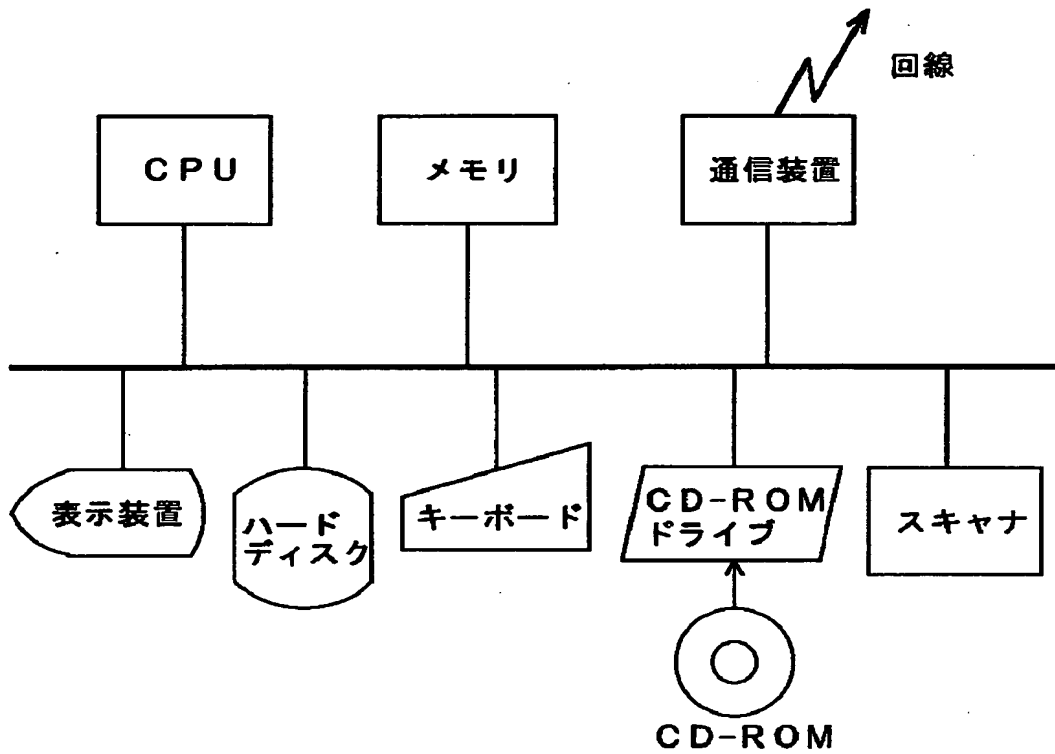


(c) 仮想境界点列



(d) 処理結果

【図51】



【書類名】 要約書

【要約】

【課題】 近接あるいは接触文字列を適切に切り出すと共に、処理時間の増大を抑制しつつ、認識精度を向上させる。

【解決手段】 入力画像データ（１）から断面系列グラフを抽出し（２）、断面系列グラフを管理するための要素タグを生成する（３）。断面系列の特異領域に対して仮想境界点列を生成する（４）。要素タグを組み合わせて文字候補を生成し、文字候補に必要な仮想境界点列を補って一文字を認識し（７）、一文字認識された文字候補間にリンクを生成し、リンク間にパスを生成し（６）、最適なパスを選択して文字列の認識結果を出力する。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [000006747]

1. 変更年月日 1990年 8月24日
[変更理由] 新規登録
住 所 東京都大田区中馬込1丁目3番6号
氏 名 株式会社リコー